

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη ιπτάμενης πλατφόρμας τύπου
Quadcopter

Development of a Quadcopter flying platform

ΣΠΥΡΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ

A.M.: 5102

Επιβλέπων Καθηγητής
Σφακιωτάκης Μιχάλης
Αναπληρωτής Καθηγητής

© Εργαστήριο Συστημάτων Αυτόματου Ελέγχου
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Σχολή Μηχανικών • ΕΛ.ΜΕ.ΠΑ.

Απαγορεύεται η αναπαραγωγή ολόκληρου ή τμήματος του κειμένου χωρίς την έγγραφη άδεια του Εργαστηρίου.

Ευχαριστίες

Η υλοποίηση της παρούσας πτυχιακής εργασίας δεν θα ήταν εφικτή χωρίς την συμβολή συγκεκριμένων ανθρώπων που έπαιξαν καταλυτικό ρόλο με τις ιδέες γνώσεις και εμπειρίες τους. Η κεντρική διαμόρφωση του αντικειμένου της παρούσης μελέτης ανήκει στον επιβλέποντα καθηγητή Δρ. Σφακιωτάκη Μιχάλη, ο οποίος συντέλεσε καθοριστικά σε θέματα ελέγχου, στην ανάπτυξη της υπάμενης πλατφόρμας και με την αδιάλειπτη καθοδήγηση του ήρθαν οι εργασίες εις πέρας. Θα ήθελα να τον ευχαριστήσω καταρχάς για την εμπιστοσύνη που μου έδειξε σχετικά με την ανάληψη της συγκεκριμένης εργασίας, για τις γνώσεις και εμπειρίες που μου προσέφερε στο πεδίο των Συστημάτων Αυτομάτου Ελέγχου, των μικροελεγκτών και της εφαρμοσμένης έρευνας. Τον ευχαριστώ επίσης για την ευκαιρία που μου έδωσε να είμαι κατά το ακαδημαϊκό έτος 2013-2014 αμειβόμενος φοιτητής στο εργαστήριο Μικροπολογιστών και Ηλεκτρονικών Κατασκευών του Τ.Ε.Ι. Κρήτης (νυν ΕΛ.ΜΕ.ΠΑ.) του οποίου είναι υπεύθυνος και το επόμενο ακαδημαϊκό έτος να πραγματοποιήσω την πρακτική μου άσκηση στο συγκεκριμένο εργαστήριο. Θα ήθελα να ευχαριστήσω ακόμα τον κο Ιωάννη Μηναδάκη από το Τμήμα Ηλεκτρολόγων Μηχανικών για την παραχώρηση διατάξεων με στόχο την βαθμονόμηση των κινητήρων, τον Δρ. Πετούση Μάρκο, επιστημονικό Συνεργάτη του Τμήματος Μηχανολόγων Μηχανικών για την παραχώρηση της χρήσης των μηχανημάτων τριοδιάστατης εκτόπωσης, τον Δρ. Εμμανουήλ Καβουσανό από το Τμήμα Μηχανολόγων Μηχανικών για την παραχώρηση της τηλεκατεύθυνσης και τον Δρ. Ιωάννη Φασουλά από το Τμήμα Μηχανολόγων Μηχανικών για τις ιδέες και τις συμβουλές του. Θα ήθελα να ευχαριστήσω ιδιαίτερος τον Νεκτάριο Σπυριδάκη, Επιστημονικό Συνεργάτη του Τμήματος Ηλεκτρολογίας για την βοήθεια, τις γνώσεις που μου μετέφερε και τις ιδέες που συζητήσαμε. Επίσης θα ήθελα να ευχαριστήσω τον Συλλιγνάκη Ιωάννη, Επιστημονικό Συνεργάτη του Τμήματος Ηλεκτρολόγων Μηχανικών για τις γνώσεις του πάνω στους κινητήρες και τις συμβουλές που μου έδωσε. Τέλος, ευχαριστώ θερμά την Ρόζα Γκλίβα, μεταπτυχιακή φοιτήτρια (και νυν απόφοιτος) του ΔΠΜΣ «Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής» για την συμβολή και τις ιδέες της σε κάθε μηχανολογικό τμήμα της συγκεκριμένης πλατφόρμας.

Κλείνοντας θα ήθελα να ευχαριστήσω την οικογένειά μου και όλα τα άτομα εκείνα, τα οποία με στήριξαν καθ' όλη την διάρκεια των σπουδών μου.

Περίληψη

Τα τελευταία χρόνια με τη διάδοση και τις ποικίλες εφαρμογές που έχουν τα μη επανδρωμένα οχήματα (UAV) έχουν μπει για τα καλά στη ζωή μας. Η τεχνολογία ευπρόσιτη πλέον για να κατασκευαστεί ένα ρομπότ οικονομικά, βοηθά στην ευρέως έρευνα και ανάπτυξη εφαρμογών των συγκεκριμένων οχημάτων. Στόχος της πτυχιακής είναι η κατασκευή και ο έλεγχος υπάμενης πλατφόρμας με τέσσερις έλικες (Quadcopter) με βαθμονομημένους κινητήρες. Σκοπός της βαθμονόμησης είναι η βελτίωση της λειτουργίας των κινητήρων, με στόχο την όμοια λειτουργία των κινητήρων για αύξηση του χρόνου πτήσης και ομαλότερη πτήση.

Στο κεφάλαιο 1 γίνεται μια αναφορά στα μη επανδρωμένα οχήματα (UAV), παρουσιάζονται διάφορες εμπορικές πλατφόρμες tricopters, quadcopters, hexacopters, octocopters και γίνεται μια αναφορά στο τρόπο με τον οποίο κινείται η πλατφόρμα στο χώρο. Στο κεφάλαιο 2 παρουσιάζεται αναλυτικά η πλατφόρμα που αναπτύχθηκε με τα επιμέρους πειράματα, τμήματα της και τα χαρακτηριστικά τους. Το κεφάλαιο 3 πραγματεύεται τις στρατηγικές ελέγχου που πραγματοποιήθηκαν, για την επίτευξη του ελέγχου γωνιακής θέσης στους παράλληλους με το έδαφος άξονες. Τέλος στο κεφάλαιο 4 παρουσιάζονται τα συμπεράσματα, τα σχόλια και οι μελλοντικές αναβαθμίσεις που μπορούν να γίνουν πάνω στη πλατφόρμα.

Από την ανάπτυξη της συγκεκριμένης υπάμενης πλατφόρμας διαφαίνεται η ανάγκη για περαιτέρω ανάπτυξη της, μέσω της προσθήκης περισσότερων αισθητήρων και ελεγκτών, η πτήση θα είναι εφικτή από οποιοδήποτε χρήστη, χωρίς να υπάρχει κίνδυνος τραυματισμού ατόμου ή πτώση της πλατφόρμας. Η χρήση μαγνητόμετρου βοηθάει στην σταθεροποίηση της διεύθυνσης και με τη χρήση του βαρόμετρου επιτυγχάνεται η πτήση σε επιθυμητό υψόμετρο. Τέλος, με τη τοποθέτηση GPS στη πλατφόρμα ο χρήστης έχει τη δυνατότητα να ελέγχει τη γεωγραφική θέση και ταχύτητα της υπάμενης πλατφόρμας.

Summary

The last years with the spread and variety of applications the unmanned aerial vehicles (UAV) entered our lives for good. The technology to make a robot now is more reachable and cheaper, so this helps in research and applications of these specific vehicles. Target of this project is the manufacture and control of an aerial platform with four propellers and calibrated motors. The use of the calibration is to upturn of the motor operation so there can be smoother and longer flight.

In chapter 1 reference is made of the unmanned vehicles, also some types of commercial platforms are presented and lastly a reference is made of how the platform moves into space. In chapter 2 is presented thoroughly the platform that was manufactured along with each component and its characteristics and the experiments that had to be done. Chapter 3 deals with the strategies of control that were done, so control of angular position could be achieved on pitch and roll axis. Lastly in chapter 4 are presented the conclusions, the comments and future upgrades that can be made on the platform.

From this project the conclusion is that further research has to be made. With the addition of more sensors and controllers a safe flight can be achieved by anyone, without the fear of someone getting injured or the crash of the platform. With the use of a magnetometer orientation can be achieved, also adding a barometer the platform will be able to maintain a certain height and lastly with the addition of a GPS the geographic position and speed can be achieved.

Περιεχόμενα

1	Εισαγωγικά Στοιχεία.....	9
1.1	Μη επανδρωμένα εναέρια οχήματα (UAVs).....	9
1.2	Είδη ιπτάμενων πλατφόρμων	10
1.2.1	Τρικόπτερα	10
1.2.2	Τετρακόπτερα.....	11
1.2.3	Εξακόπτερα	13
1.2.4	Οκτακόπτερα	15
1.3	Κίνηση της ιπτάμενης πλατφόρμας στο χώρο.....	17
1.3.1	Απογείωση/προσγείωση/αλλαγή ύψους	17
1.3.2	Κίνηση μπροστά/πίσω (X άξονας)	18
1.3.3	Κίνηση αριστερά/δεξιά (Y άξονας).....	19
1.3.4	Αλλαγή κατεύθυνσης (περιστροφή γύρω από τον Z άξονα)	20
2	ΑΝΑΠΤΥΞΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ	22
2.1	Εισαγωγή	22
2.2	Επισκόπηση πλατφόρμας.....	22
2.3	Κινητήρες Brushless.....	24
2.3.1	Γενικές πληροφορίες	24
2.3.2	Οδήγηση Brushless κινητήρων.....	24
2.3.3	Βαθμονόμηση λειτουργίας DC κινητήρα BRUSHLESS.....	26
2.4	Μπαταρία	30
2.4.1	Γενικές πληροφορίες	30
2.4.2	Φόρτιση μπαταρίας	31
2.5	Έλικες	32
2.5.1	Γενικές πληροφορίες για τις έλικες	32
2.5.2	Εξισορρόπηση έλικας.....	32
2.6	Πλαίσιο Ιπτάμενης πλατφόρμας.....	34
2.6.1	Πλαίσιο.....	34
2.6.2	Αντικραδασμικά	34
2.7	Τηλεκατεύθυνση	37
2.7.1	Αποτελούμενα μέρη της τηλεκατεύθυνσης	37
2.8	Μικροελεγκτής.....	39
2.9	Πλακέτα επέκτασης μικροελεγκτή.....	41
2.10	MPU-6050 6DOF (Γυροσκόπιο, Επιταχυνσιόμετρο)	44
2.10.1	Γυροσκόπιο	44
2.10.2	Επιταχυνσιόμετρο	45
2.10.3	Αδρανειακή Μονάδα Μέτρησης	46
3	Στρατηγικές Ελέγχου της ιπτάμενης πλατφόρμας.....	51
3.1	Εισαγωγή στον έλεγχο της ιπτάμενης πλατφόρμας	51
3.2	Στρατηγικές ελέγχου της ιπτάμενης πλατφόρμας.....	51
3.2.1	Έλεγχος στους X, Y άξονες.....	51

3.2.2 Έλεγχος στον Z άξονα.....	65
3.3 Ροή του προγράμματος ελέγχου.....	70
4 Σχόλια, Συμπεράσματα και Προτάσεις για Μελλοντική Εξέλιξη.....	73
4.1 Σχόλια, Συμπεράσματα.....	73
4.2 Προτάσεις για Μελλοντική Εξέλιξη.....	73

Σχήματα

Σχήμα 1 Διάταξη κινήτρων ενός τρικόπτερου	10
Σχήμα 2 Shrediquette DLX	11
Σχήμα 3 Tricopter V3.5	11
Σχήμα 4 Η διάταξη Σταυρού (QUAD+) και η Χιαστή διάταξη (QUAD X) τετρακόπτερων.	12
Σχήμα 5 Phantom Vision 2	12
Σχήμα 6 DJI Inspire 1	13
Σχήμα 7 Η διάταξη Σταυρού (HEXA+) και η Χιαστή διάταξη (HEXA X) εξακόπτερων.	13
Σχήμα 8 S550 Hexacopter	14
Σχήμα 9 Hextreme FPV Hexacopter	14
Σχήμα 10 Spyder Extreme FPV Hexacopter	15
Σχήμα 11 Η διάταξη Σταυρού (OCTA+) και η Χιαστή διάταξη (OCTA X) Οκτακόπτερων	15
Σχήμα 12 Spreading Wings S1000	16
Σχήμα 13 Skyjib	16
Σχήμα 14 Πλαίσιο Γης, πλαίσιο σώματος και ιπτάμενη πλατφόρμα	17
Σχήμα 15 Πάνω σχήμα μείωση στροφών (προσγείωση), κάτω σχήμα αύξηση στροφών(απογείωση)	18
Σχήμα 16 Πάνω σχήμα κίνηση προς τα πίσω, κάτω σχήμα κίνηση προς τα μπρος	19
Σχήμα 17 Πάνω σχήμα κίνηση δεξιά, κάτω σχήμα κίνηση αριστερά	20
Σχήμα 18 Πάνω σχήμα κίνηση δεξιόστροφα, κάτω σχήμα κίνηση αριστερόστροφα	21
Σχήμα 19 Η πλατφόρμα που αναπτύχθηκε.	22
Σχήμα 20 Πτήση της πλατφόρμας με περιορισμό ύψους.	23
Σχήμα 21 Συνδεσμολογία των τρανζίστορ στα αριστερά με τα τυλίγματα του κινητήρα δεξιά	25
Σχήμα 22 (α) Ηλεκτρονική μονάδα ελέγχου ταχύτητας κινητήρων brushless (electronic speed controller, ESC). (β)Τεχνικά χαρακτηριστικά της μονάδας ESC.	25
Σχήμα 23 EMAX CF2822 1200KV	26
Σχήμα 24 Η διάταξη που υλοποιήθηκε για τα πειράματα	27
Σχήμα 26 Μετρήσεις ώσης σε γραμμάρια σε γράφημα από το πείραμα 1	28
Σχήμα 28 Αφιλτράριστες μετρήσεις από πείραμα 2	29
Σχήμα 29 Γράφημα επιβεβαίωσης βαθμονόμησης από πείραμα 2	30
Σχήμα 30 Η έλικα 'ζυγισμένη' στην διάταξη.	33
Σχήμα 31 Η έλικα στην διάταξη εξισορρόπησης	33
Σχήμα 32 Πλαίσιο ιπτάμενης πλατφόρμας F450 της DJI	34
Σχήμα 33 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον X άξονα	35
Σχήμα 34 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον Y άξονα	35
Σχήμα 35 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον Z άξονα	36
Σχήμα 36 Ο κινητήρας με την έλικα τοποθετημένος στο πόδι του πλαισίου με το plexiglass και τα αντικραδασμικά	36
Σχήμα 37 Αντικραδασμικές σφαίρες	37
Σχήμα 38 Πομπός συστήματος τηλεκατεύθυνσης Spektrum DX6i	39
Σχήμα 39 Δέκτης του συστήματος της τηλεκατεύθυνσης	39

Σχήμα 40 Το αναπτυσσόμενο Arduino Mega 2560, και το λειτουργικό διάγραμμα του μικροελεγκτή	40
Σχήμα 41 Πλακέτα επέκτασης του μικροελεγκτή	42
Σχήμα 45 Διασύνδεση MPU-6050 με μικροελεγκτή	42
Σχήμα 42 Η διάταξη των 3 σημάτων ανά header για την διασύνδεση του ESC με τον μικροελεγκτή	42
Σχήμα 43 Η διάταξη των 3 σημάτων ανά header για την διασύνδεση του δέκτη της τηλεκατεύθυνσης με τον μικροελεγκτή	43
Σχήμα 44 Σήμα στο μέγιστο, στην μέση και στο ελάχιστο από τον δέκτη της τηλεκατεύθυνσης	43
Σχήμα 46 (α) γυροσκόπιο RLG (β) γυροσκόπιο HRG (γ) γυροσκόπιο FOG (δ) διάγραμμα γυροσκοπίου MEMS (ε) γυροσκόπιο 1 άξονα (MEMS)	45
Σχήμα 47 (α) Χωρητικό Επιταχυνσιόμετρο (β) Οπτικό Επιταχυνσιόμετρο (γ) Πιεζοηλεκτρικό Επιταχυνσιόμετρο (δ) Επιταχυνσιόμετρο Αντίστασης (ε) Επιταχυνσιόμετρο Φυσαλίδας Αέρα	46
Σχήμα 48 Μετρήσεις Επιταχυνσιομέτρου	47
Σχήμα 49 Μετρήσεις Γυροσκοπίου	48
Σχήμα 50 Φιλτραρισμένες Μετρήσεις	48
Σχήμα 51 MPU-6050	50
Σχήμα 52 Διάγραμμα ελεγκτή Γωνιακής ταχύτητας X άξονα	52
Σχήμα 53 Η απόκριση του συστήματος με παρεμβολές προς τις δύο κατευθύνσεις.	53
Σχήμα 54 Απόκριση γωνιακής θέσης του ελεγκτή γωνιακής ταχύτητας	53
Σχήμα 56 Απόκριση ελεγκτή γωνιακής θέσης με επιθυμητή γωνιακή ταχύτητα 10deg/sec	55
Σχήμα 58 Σήματα ελέγχου των κινητήρων για τον ελεγκτή γωνιακής ταχύτητας	56
Σχήμα 59 Διάγραμμα γωνιακής θέσης P-D ελεγκτή X άξονα	56
Σχήμα 60 Απόκριση γωνιακής θέσης P-D ελεγκτή	57
Σχήμα 61 Σήματα Ελέγχου του P-D ελεγκτή γωνιακής θέσης	58
Σχήμα 62 Απόκριση γωνιακής θέσης P-D ελεγκτή για επιθυμητή γωνιακή θέση 20 μοίρες	59
Σχήμα 63 Σήματα ελέγχου των κινητήρων	59
Σχήμα 65 Απόκριση γωνιακής θέσης PI-D ελεγκτή για επιθυμητή γωνιακή θέση τις 0 μοίρες	61
Σχήμα 66 Σήματα ελεγκτή για τους κινητήρες	62
Σχήμα 67 Απόκριση γωνιακής θέσης PI-D ελεγκτή με επιθυμητή γωνιακή θέση τις 15 μοίρες	63
Σχήμα 68 Σήματα του ελεγκτή για τους κινητήρες	63
Σχήμα 69 Απόκριση γωνιακής θέσης του PI-D ελεγκτή με την χρήση τηλεκατεύθυνσης	64
Σχήμα 70 Σήματα ελέγχου των κινητήρων από τον ελεγκτή	65
Σχήμα 71 Διάγραμμα ελέγχου γωνιακής ταχύτητας με P-D ελεγκτή	66
Σχήμα 72 Απόκριση γωνιακής ταχύτητας του P-D ελεγκτή για επιθυμητή γωνιακή ταχύτητα 0	66
Σχήμα 73 Σήματα ελέγχου κινητήρων 7 και 3	67
Σχήμα 74 Σήματα ελέγχου κινητήρων 6 και 8	67
Σχήμα 75 Διάγραμμα ελέγχου PI-D ελεγκτή	68
Σχήμα 77 Σήματα ελέγχου των κινητήρων M7 και M3	69
Σχήμα 78 Σήματα ελέγχου των κινητήρων M6 και M8	70
Σχήμα 79 Διάγραμμα ροής βρόγχου ελέγχου	72

Πίνακες

Πίνακας 1 Ακολουθία οδήγησης τρανζίστορ	25
Πίνακας 2 Πολύνομα ανά μονάδα έλικα/κινητήρα	30
Πίνακας 3 Χαρακτηριστικά μπαταριών.....	31
Πίνακας 4 Αντιστοίχιση σημάτων PWM με κινητήρες	41
Πίνακας 5 Κέρδη ελεγκτών	70

1 Εισαγωγικά Στοιχεία

1.1 Μη επανδρωμένα εναέρια οχήματα (UAVs)

Τα μη επανδρωμένα εναέρια οχήματα (Unmanned Aerial Vehicles, UAVs) είναι υπτάμενα σκάφη χωρίς ανθρώπινο δυναμικό. Χρησιμοποιούν τις αεροδυναμικές δυνάμεις για να επιτύχουν την άνωση του σκάφους. Μπορούν να πετάξουν αυτόνομα ή και με τηλεχειρισμό, είναι αναλώσιμα, ανακτήσιμα και μπορούν να μεταφέρουν κάποιο φορτίο.

Οι πιο εξελιγμένες εκδόσεις μπορούν να χρησιμοποιούν συστήματα αυτομάτου ελέγχου σε συνδυασμό με συστήματα πλοήγησης, με σκοπό την εκτέλεση κάποιων λειτουργιών, όπως για παράδειγμα είναι η πτήση με σταθερή ταχύτητα, η σταθεροποίηση της πορείας πτήσης και η διατήρηση σταθερής θέσης.

Τα πρώτα UAVs ήταν απλώς τηλεχειριζόμενα αεροσκάφη ενώ πλέον υπάρχει μια ευρεία ποικιλία σε σχήματα, μεγέθη και χαρακτηριστικά.

Αναφορικά Τα UAVs έχουν δύο κατηγορίες:

- αυτά που ελέγχονται εξ'αποστάσεως από τον χειριστή
- αυτά που πετούν αυτόνομα με βάση μια προγραμματισμένη διαδρομή

Οι πρώτες εφαρμογές των UAVs εξυπηρετούσαν στρατιωτικούς σκοπούς.

Ειδικότερα εφαρμόζονταν πρώτον, σαν στόχος και παραπλάνηση δημιουργώντας ένα ψεύτικο στόχο στον εχθρό. Δεύτερον, για αναγνώριση παρέχοντας πληροφορίες για μια περιοχή. Τέλος, σε μάχες για εκτέλεση αποστολών υψίστου κινδύνου.

Με την πάροδο των χρόνων η χρήση των UAVs έχει επεκταθεί και σε άλλες σε εφαρμογές[Α1], όπως:

Κινηματογράφηση - λήψεις σκηνών χωρίς γερανούς και ελικόπτερα, με σημαντικά μειωμένο κόστος.

Πυρόσβεση

Μεταφορές - παράδοση μικρών πακέτων

Αεροφωτογραφία - λήψη εικόνων από ύψος για χρήση σε ταξιδιωτικούς οδηγούς

Έλεγχος κτιρίων και άλλων κατασκευών - γρηγορότερος έλεγχος και ευκολότερη πρόσβαση σε δύσβατα σημεία

Διάσωση - ταχύτερη κάλυψη περιοχών θαλάσσιων ή ορεινών για εύρεση χαμένων ατόμων

Βιντεοσκοπική κάλυψη γεγονότων - εκπομπών, δεξιώσεων κ.α.

1.2 Είδη ιπτάμενων πλατφόρμων

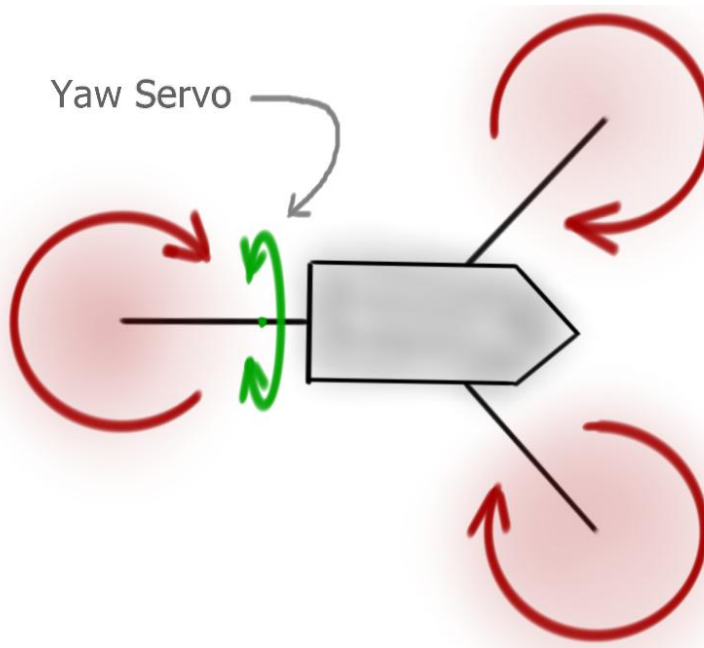
Οι ιπτάμενες πλατφόρμες χωρίζονται στις ακόλουθες 4 βασικές κατηγορίες, ανάλογα με τον αριθμό των κινητήρων που χρησιμοποιούν:

- Τρικόπτερα - χρήση τριών κινητήρων για την πτήση
- Τετρακόπτερα - χρήση τεσσάρων κινητήρων για την πτήση
- Εξακόπτερα - χρήση έξι κινητήρων για την πτήση
- Οκτακόπτερα - χρήση οχτώ κινητήρων για την πτήση

Στην κάθε κατηγορία υπάγονται πλατφόρμες: είτε τοποποιημένες είτε χειροποίητες σε διάφορα μεγέθη και χαρακτηριστικά.

Τρικόπτερα

Τα τρικόπτερα διαθέτουν τρεις κινητήρες και ισάριθμους έλικες καθώς και έναν σερβομηχανισμό θέσης (συνήθως τύπου R/C servo). Ο οποίος ελέγχει την γωνιακή θέση του κινητήρα της ουράς, για την κίνηση γύρω από τον κάθετο άξονα του σκάφους, όπως απεικονίζεται στο ακόλουθο σχήμα.



Σχήμα 1 Διάταξη κινητήρων ενός τρικόπτερου

Οι δύο εμπρόσθιοι έλικες στρέφονται αντίστροφα μεταξύ τους, ενώ ο ουραίος στρέφεται με την ίδια φορά με έναν από τους δύο εξ'αυτών. Με την χρήση του servo, δημιουργείται

αντίθετη ροπή από αυτήν που δημιουργεί ο κινητήρας της ουράς, με αποτέλεσμα να μην υπάρχει στροφή της πλατφόρμας στον Z άξονα. Αλλάζοντας τις στροφές του κινητήρα της ουράς και την γωνία του ως προς τον Y άξονα έχουμε έλεγχο στον Z άξονα.

Στα σχήματα που ακολουθούν εικονίζονται ορισμένα τρικόπτερα του εμπορίου.



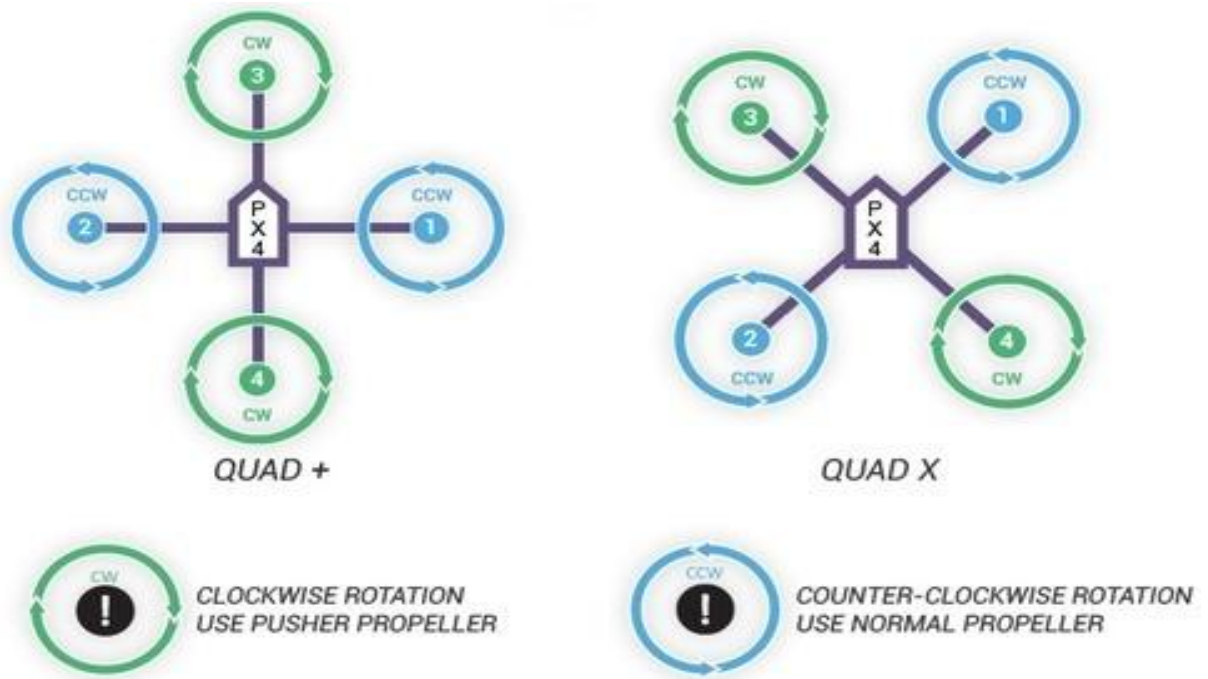
Σχήμα 2 Shrediquette DLX



Σχήμα 3 Tricopter V3.5

Τετρακόπτερα

Τα τετρακόπτερα (quadcopters) διαθέτουν τέσσερις συνολικά μονάδες έλικα/κινητήρα. Χωρίζονται σε δύο διατάξεις ανάλογα με την κίνηση του οχήματος, οι οποίες είναι η **Χιαστή** διάταξη και η διάταξη **Σταυρού**. Όπως φαίνεται στο παρακάτω σχήμα, ανεξαρτήτως της διάταξης του οχήματος, οι κινητήρες M1 και M2 στρέφονται αριστερόστροφα και οι κινητήρες M3 και M4 στρέφονται δεξιόστροφα. Με αυτόν τον τρόπο εξισορροπούνται οι δυνάμεις περιστροφής που δημιουργεί ο κάθε κινητήρας γύρω από τον κάθετο άξονα (Z άξονας) του οχήματος.



Σχήμα 4 Η διάταξη Σταυρού (QUAD+) και η Χιαστή διάταξη (QUAD X) τετρακόπτερων.

Ειδικότερα, στην διάταξη Σταυρού ο κάθε κινητήρας λαμβάνει εντολές μόνο για δύο άξονες, οι οποίες είναι η μία για τον άξονα στον οποίο ανήκει και η δεύτερη για τον Z άξονα. Αντίθετα, στη Χιαστή διάταξη απαιτείται πιο σύνθετο σχήμα ελέγχου καθώς λαμβάνει εντολές και για τους τρεις άξονες, παρέχει όμως καλύτερο έλεγχο της πλατφόρμας.

Χαρακτηριστικά παραδείγματα quadcopters εικονίζονται παρακάτω.



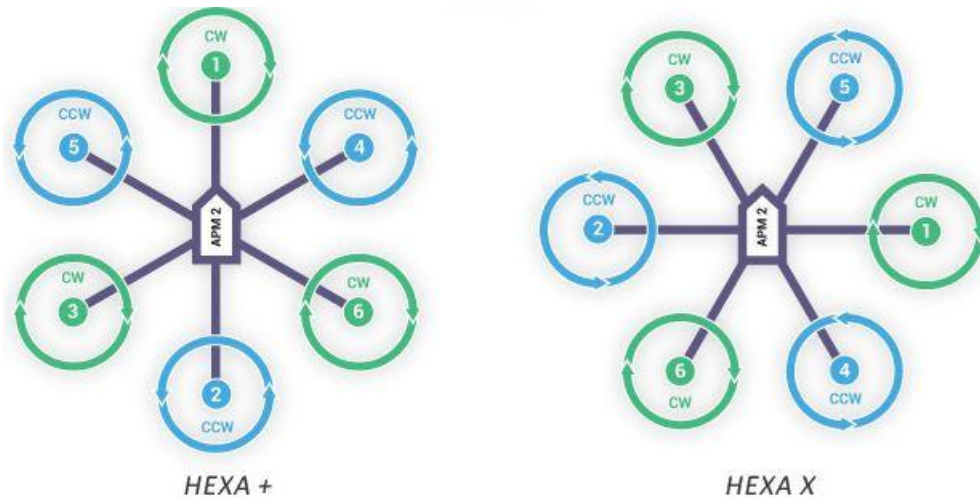
Σχήμα 5 Phantom Vision 2



Σχήμα 6 DJI Inspire 1

Εξακόπτερα

Τα εξακόπτερα διαθέτουν έξι μονάδες έλικα/κινητήρα σε δύο διατάξεις ανάλογα με την κατεύθυνση του οχήματος, οι οποίες, κατ' αντιστοιχία με τα τετρακόπτερα είναι η **Χιαστή** διάταξη και η διάταξη **Σταυρού**, οι οποίες εικονίζονται στο παρακάτω σχήμα.



Σχήμα 7 Η διάταξη Σταυρού (HEXA+) και η Χιαστή διάταξη (HEXA X) εξακόπτερον.

Μια βασική διαφορά μεταξύ του τετρακόπτερου και του εξακόπτερου είναι ότι στο πρώτο οι απέναντι κινητήρες στρέφονται με την ίδια φορά για την εξισορρόπηση των δυνάμεων στον Z άξονα, ενώ στο εξακόπτερο οι απέναντι κινητήρες στρέφονται με αντίθετη φορά. Συγκεκριμένα, οι κινητήρες M1,M3,M6 στρέφονται δεξιόστροφα και οι κινητήρες M2,M4,M5 αριστερόστροφα.

Επίσης στο εξακόπτερο οι ελεγκτές είναι πιο περίπλοκοι σχετικά με τις εντολές κίνησης, διότι η θέση του εκάστοτε κινητήρα στην πλατφόρμα και τη γωνία που σχηματίζει με τον

ελεγχόμενο άξονα είναι καθοριστική. Επιπλέον, για κάθε διάταξη υπάρχουν διάφοροι τρόποι ελέγχου της πλατφόρμας.

Λόγω της χρήσης περισσότερων κινητήρων για την κίνηση τους στο χώρο, τα εξακόπτερα είναι γενικά πιο ευκίνητα, σταθερά και με μεγαλύτερες δυνατότητες ανάληψης ωφέλιμου φορτίου. Τέλος τα εξακόπτερα λόγω μεγαλύτερης σταθερότητας σε σχέση με τα τετρακόπτερα χρησιμοποιούνται σε εφαρμογές βιντεοσκόπησης και συστημάτων FPV(first person view) όπου με χρήση κάμερας γίνεται το πιλοτάρισμα του UAV.

Στα ακόλουθα σχήματα εικονίζονται τρία εξακόπτερα του εμπορίου:



Σχήμα 8 S550 Hexacopter



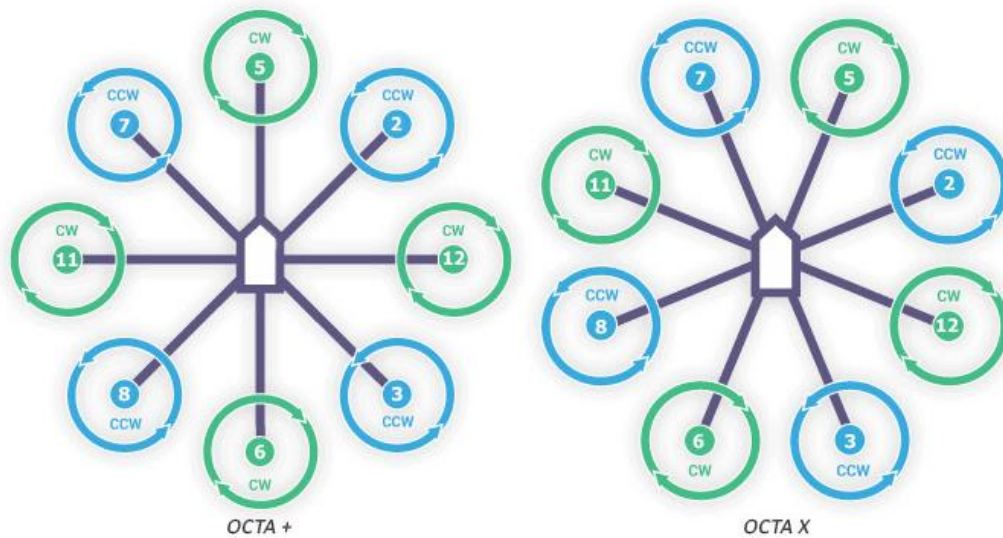
Σχήμα 9 Hextreme FPV Hexacopter



Σχήμα 10 Spyder Extreme FPV Hexacopter

Οκτακόπτερα

Τα οκτακόπτερα διαθέτουν οκτώ μονάδες έλικα/κινητήρα. Όπως φαίνεται στο παρακάτω σχήμα, και εδώ υπάρχουν οι δύο παραλλαγές στη διάταξη των κινητήρων, η **Χιαστή** διάταξη και η διάταξη **Σταυρού**.



Σχήμα 11 Η διάταξη Σταυρού (OCTA+) και η Χιαστή διάταξη (OCTA X) Οκτακόπτερον

Στο οκτακόπτερο λόγω της χρήσης οκτώ κινητήρων και της συσχέτισης με το τετρακόπτερο οι απέναντι κινητήρες στρέφονται με την ίδια φορά. Συγκεκριμένα, οι κινητήρες M11, M12, M5, M6 στρέφονται δεξιόστροφα, και οι κινητήρες M2, M3, M7, και M8 αριστερόστροφα.

Ο έλεγχος στα οκτακόπτερα είναι αναλόγως με τον έλεγχο των εξακόπτερων με την διαφορά στον έλεγχο που αφορά τον Z άξονα.

Τα οκτακόπτερα επίσης σε σχέση με τα τετρακόπτερα και τα εξακόπτερα είναι ακόμα πιο σταθερά, άρα και πιο αξιόπιστα στη χρήση κάμερας, είτε για βιντεοσκόπηση είτε για πτήση σε πρώτο πλάνο (FPV), έχουν μεγαλύτερη ισχύ συνεπώς χρησιμοποιούνται και στις μεταφορές φορτίων.

Στα ακόλουθα σχήματα εικονίζονται δύο οκτακόπτερα του εμπορίου.



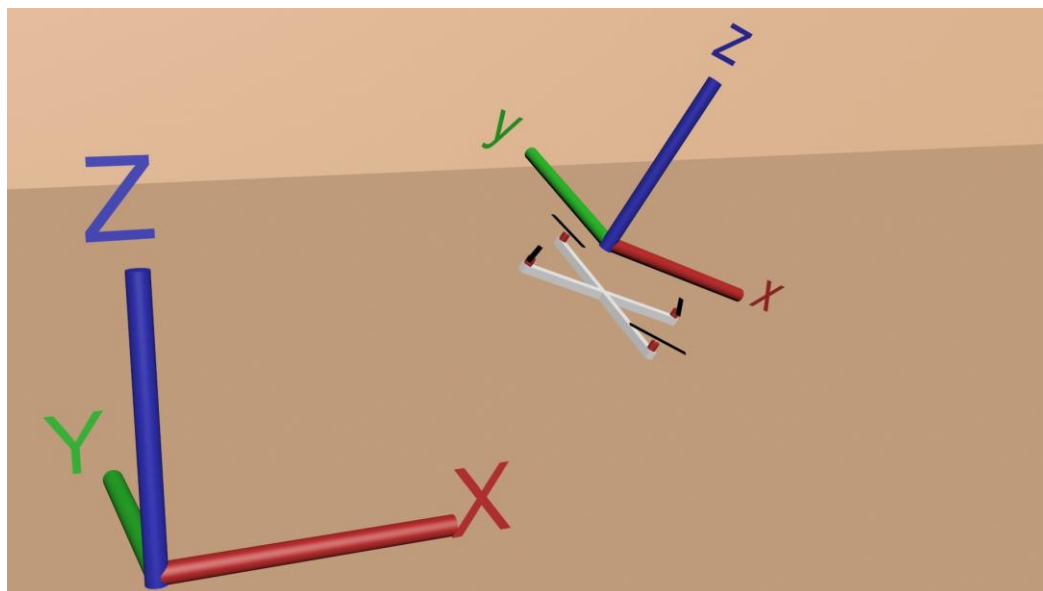
Σχήμα 12 Spreading Wings S1000



Σχήμα 13 Skyjib

1.3 Κίνηση της ιπτάμενης πλατφόρμας στο χώρο

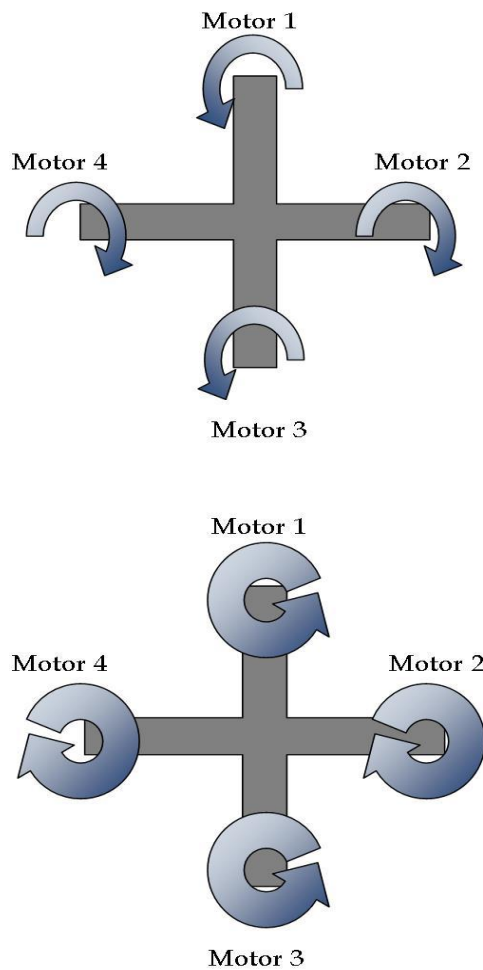
Για την κατανόηση της κίνησης της πλατφόρμας πρέπει να ορίσουμε τον χώρο μέσα στον οποίο κινείται. Ο χώρος ορίζεται με πλαίσιο αναφοράς τη Γη στις 3 διαστάσεις (X,Y,Z). Για την κίνηση της πλατφόρμας ορίζεται πλαίσιο μάζας της πλατφόρμας (x,y,z). Η IMU συγκρίνοντας τα δύο πλαίσια, ορίζει την γωνιακή θέση της πλατφόρμας σε σχέση με τη Γη για χρήση της στον ελεγκτή θέσης. Στο ακόλουθο σχήμα (σχήμα 14) παρουσιάζονται τα δύο πλαίσια και η ιπτάμενη πλατφόρμα. Η κίνηση της ιπτάμενης πλατφόρμας στον χώρο εξαρτάται από τον σχεδιασμό του ελεγκτή και τη διάταξη των κινητήρων. Καθώς η συγκεκριμένη πλατφόρμα, που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής είναι τετρακόπτερο σε διάταξη Σταυρού, η περιγραφή που ακολουθεί για τις βασικές αρχές κίνησης δημοσιεύτηκε από την εξής διεθνή βιβλιογραφία[A1] και αναφέρεται σε αυτήν την περίπτωση.



Σχήμα 14 Πλαίσιο Γης, πλαίσιο σώματος και ιπτάμενη πλατφόρμα

Απογείωση/προσγείωση/αλλαγή ύψους

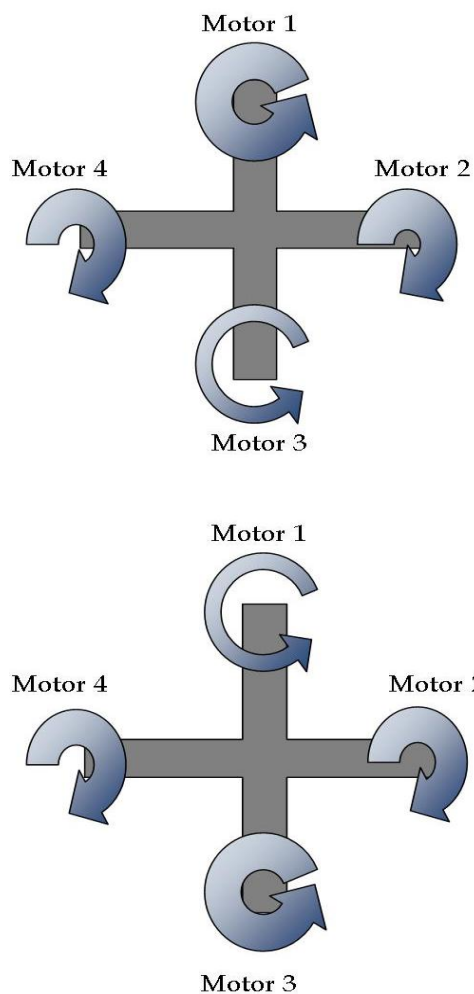
Για την απογείωση, προσγείωση και αλλαγή ύψους απαιτείται η ρύθμιση των στροφών και των τεσσάρων κινητήρων. Εφόσον απαιτείται η αμιγώς κάθετη κίνηση του οχήματος, στις περιπτώσεις αυτές θα πρέπει, οι κινητήρες να παράγουν δυνάμεις ίσες και με την ίδια κατεύθυνση. Η κίνηση καθορίζεται ανάλογα με την επιθυμητή κατεύθυνση (προς τα πάνω ή προς τα κάτω). Για την απογείωση/αύξηση ύψους χρειάζεται η αύξηση των στροφών των κινητήρων, στο σημείο που η δύναμη που παράγουν να υπερνικά την βαρύτητα για την απογείωση. Αντίθετα για μείωση ύψους ή προσγείωση απαιτείται η ελάττωση των στροφών. Στο ακόλουθο σχήμα φαίνονται οι δυνάμεις ανά κινητήρα:



Σχήμα 15 Πάνω σχήμα μείωση στροφών (προσγείωση), κάτω σχήμα αύξηση στροφών(απογείωση)

Κίνηση μπροστά/πίσω (X άξονας)

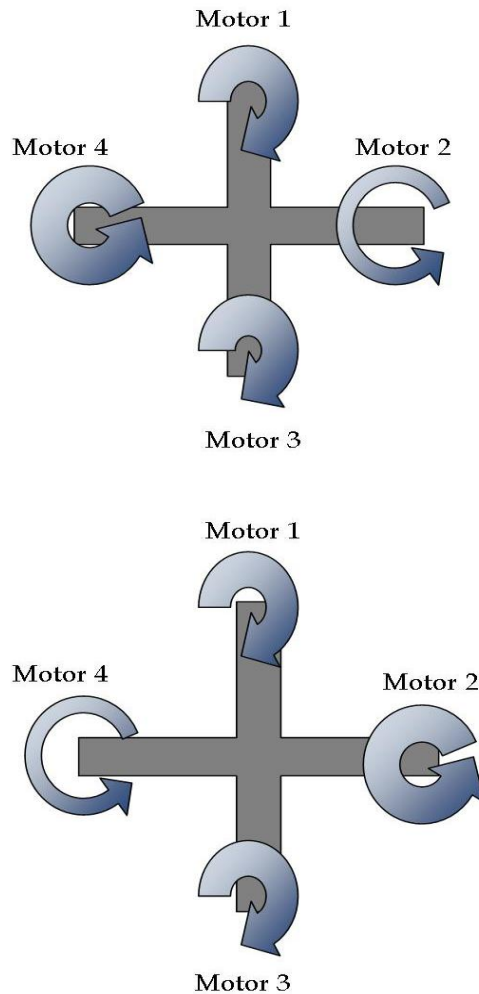
Για την κίνηση μπροστά/πίσω απαιτείται η αλλαγή των στροφών των κινητήρων M1 και M3. Συγκεκριμένα, για την κίνηση μπροστά μειώνονται οι στροφές του κινητήρα M1 και αυξάνονται οι στροφές του κινητήρα M3 κατά τον ίδιο βαθμό, όπως εικονίζεται στο επόμενο σχήμα. Αποτέλεσμα της αυξομείωσης αυτής είναι, η πλατφόρμα να παρεκκλίνει από την οριζόντια, στην θέση που ζητείται (μεταβολή της γωνίας πρόνευσης, pitch angle). Η συνισταμένη των δυνάμεων στον X άξονα, έχει διεύθυνση προς τα μπροστά και αναγκάζει την πλατφόρμα να κινείται μπροστά. Η ταχύτητα κίνησης προς τα μπροστά αυξάνεται όσο μεγαλώνει η γωνία πρόνευσης. Σημειώνεται ότι, η αλλαγή της συνισταμένης των δυνάμεων σε έναν άξονα, έχει ως αποτέλεσμα στον κατακόρυφο να μειώνεται η κάθετη δύναμη ανάλογα με το μέγεθος της γωνίας, με συνέπεια να χάνει ύψος η πλατφόρμα. Το πρόβλημα αυτό αντιμετωπίζεται με την αύξηση της συνολικής δύναμης. Για την κίνηση προς τα πίσω, η αυξομείωση των στροφών για τους κινητήρες M1 και M3 είναι η αντίθετη απ' ό τι για κίνηση προς τα μπροστά.



Σχήμα 16 Πάνω σχήμα κίνηση προς τα πίσω, κάτω σχήμα κίνηση προς τα μπρος

Κίνηση αριστερά/δεξιά (Y άξονας)

Για την κίνηση αριστερά δεξιά απαιτείται η αλλαγή των στροφών των κινητήρων M2 και M4, με τρόπο αντίστοιχο αυτού για την κίνηση κατά τον X άξονα. Συγκεκριμένα, για κίνηση προς τα αριστερά απαιτείται αύξηση των στροφών του κινητήρα M2 και μείωση των στροφών του κινητήρα M4. Το ακριβώς αντίστροφο συμβαίνει για κίνηση προς τα δεξιά. Η αλλαγή των στροφών των κινητήρων M2 και M4 φαίνεται στο ακόλουθο σχήμα:

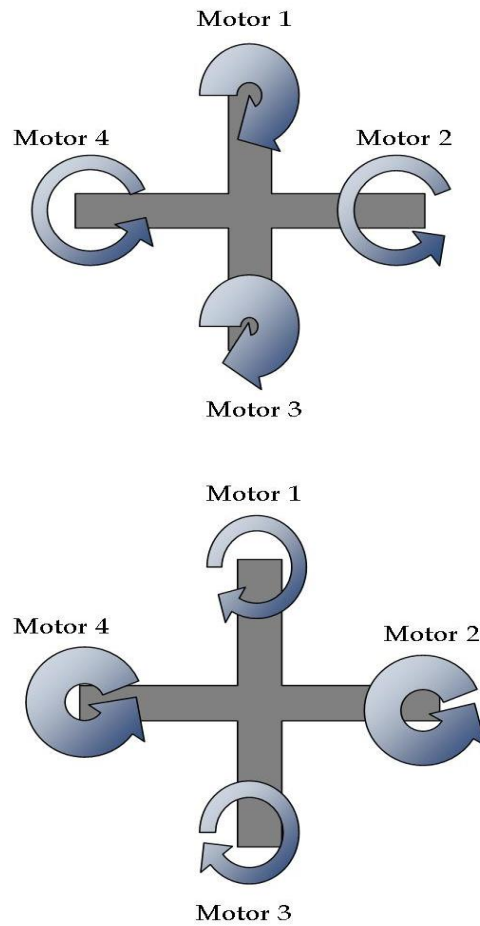


Σχήμα 17 Πάνω σχήμα κίνηση δεξιά, κάτω σχήμα κίνηση αριστερά

Αλλαγή κατεύθυνσης (περιστροφή γύρω από τον Z άξονα)

Κατεύθυνση ορίζουμε την γωνία στον Z άξονα, όπου η πλατφόρμα κοιτάζει μπροστά. Ο κινητήρας M1 θεωρείται ότι κοιτάζει μπροστά, ο κινητήρας M3 κοιτάζει πίσω, ο κινητήρας M2 αριστερά και ο κινητήρας M4 δεξιά. Η περιστροφή γύρω από τον κάθετο άξονα του οχήματος, επιτυγχάνεται με αλλαγή των στροφών και για τους τέσσερις κινητήρες. Για την επίτευξη αλλαγής κατεύθυνσης αριστερόστροφα ή δεξιόστροφα οι στροφές των κινητήρων αυξάνουν και μειώνουν ταυτόχρονα ανά ζευγάρια. Τα ζευγάρια είναι πάντοτε οι κινητήρες που στρέφονται προς την ίδια φορά. Το κάθε ζευγάρι παράγει δύναμη, που θέλει να στρέψει την πλατφόρμα προς την φορά που στρίβει το εκάστοτε ζευγάρι. Με ίσες στροφές και στα δύο ζευγάρια, επιτυγχάνεται ισορροπία στην κατεύθυνση. Έτσι ανάλογα με την αυξομείωση των ζευγαριών, δύναται να ορίσουμε την επιθυμητή κατεύθυνση. Συγκεκριμένα, οι κινητήρες M1 και M3 που στρέφονται δεξιόστροφα αυξάνουν ή μειώνουν τις στροφές τους σε σχέση με τους κινητήρες M2 και M4 που στρέφονται προς την αντίθετη φορά. Πχ αν ο κινητήρας M1 κοιτάζει στον βορρά και θέλουμε να κοιτάζει στην ανατολή, θα αυξήσουμε τις στροφές των M1,M3 και θα μειώσουμε στον ίδιο βαθμό τις στροφές στους κινητήρες M2,M4.

Όταν ο κινητήρας M1 κοιτάζει πλέον στην ανατολή θα επαναφέρουμε τις στροφές σε ισορροπία για να διατηρήσουμε την κατεύθυνση. Η αυξομείωση των στροφών των ζευγαριών εικονίζεται στο ακόλουθο σχήμα:



Σχήμα 18 Πάνω σχήμα κίνηση δεξιόστροφα, κάτω σχήμα κίνηση αριστερόστροφα

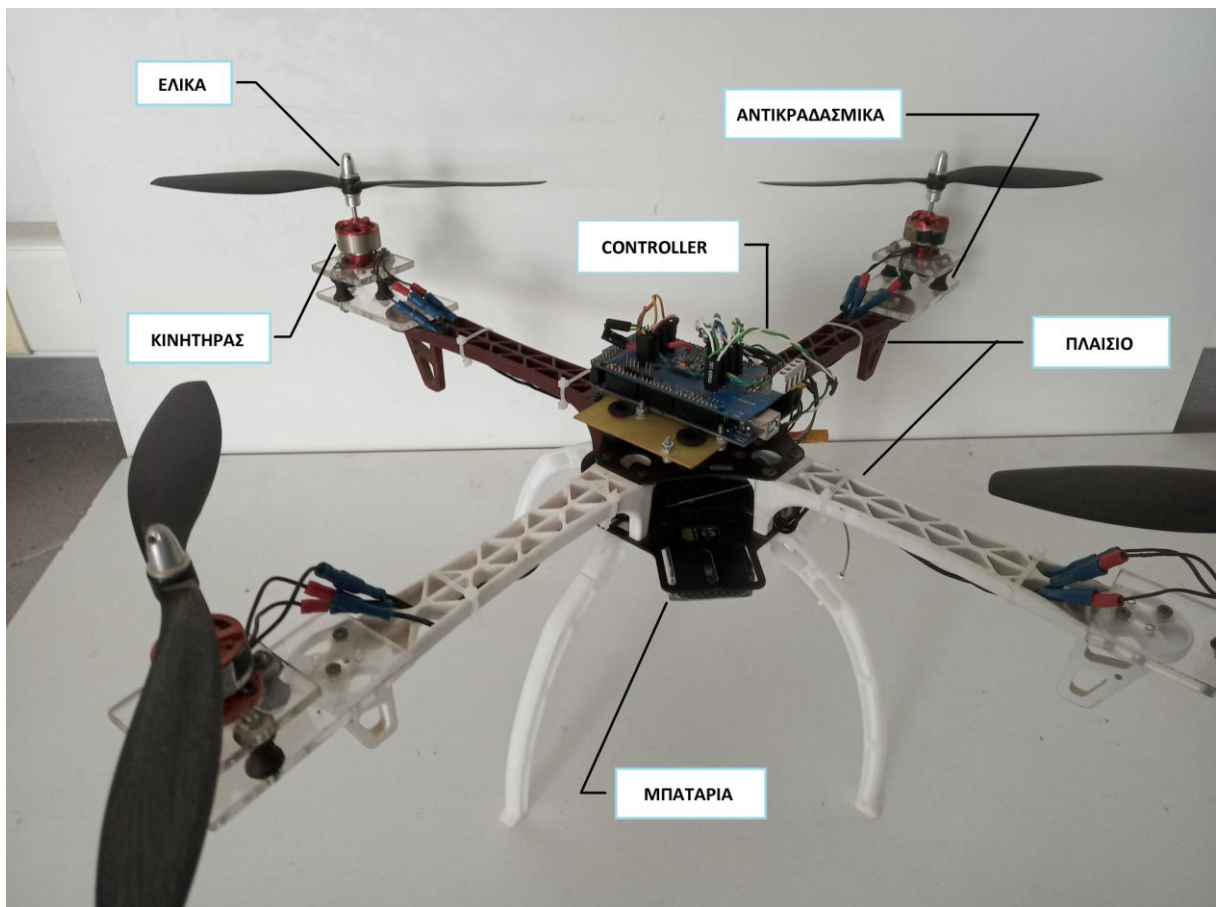
2 ΑΝΑΠΤΥΞΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζονται οι γενικές πληροφορίες και τα χαρακτηριστικά του κάθε εξαρτήματος που χρησιμοποιήθηκε για την ανάπτυξη της υπάμενης πλατφόρμας.

2.2 Επισκόπηση πλατφόρμας

Η πλατφόρμα αναπτύχθηκε με σκοπό, την βελτιστοποίηση πτήσης με χρήση βαθμονομημένων κινητήρων. Η βαθμονόμηση κάθε κινητήρα μπορεί να επιφέρει ως ένα βαθμό όμοια λειτουργία όλων των κινητήρων, αποτέλεσμα είναι η μείωση κατανάλωσης ισχύος και η αύξηση του χρόνου πτήσης.



Σχήμα 19 Η πλατφόρμα που αναπτύχθηκε.

Οι δυνατότητες της πλατφόρμας είναι οι εξής:

- Το συνολικό βάρος που μπορεί να σηκώσει είναι 3.2kg (ωφέλιμο 1.5kg).
- Διατήρηση της επιθυμητής γωνιακής θέσης στους άξονες X, Y ακόμα και με παρεμβολές.
- Έλεγχος γωνιακής ταχύτητας και στους 3 άξονες (X,Y,Z).

Η πλατφόρμα που αναπτύχθηκε διαθέτει τα εξής εξαρτήματα :

Πλαίσιο F450 της DJI με μήκος 450mm από άκρη σε άκρη.

Κινητήρες CF2822 της EMAX στα 1200KV.

Μικροελεγκτής Arduino Mega 2560.

Έλικες 1045.

Μπαταρίες Λιθίου-Πολυμερίου 11.1V 3 κελιών

Πλακέτα επέκτασης για Arduino Mega.

Αντικραδασμικές σφαίρες.

Αντικραδασμικές βίδες.

Plexiglass σε δύο ξεχωριστές διαστάσεις.

Esc Skywalker Quatro 25Ax4.

Τηλεκατεύθυνση DX6i Spektrum με τον δέκτη της.

Σφαίρες αλουμινίου στήριξης έλικας σε άξονα 3mm.



Σχήμα 20 Πτήση της πλατφόρμας με περιορισμό ύψους.

2.3 Κινητήρες Brushless

Γενικές πληροφορίες

Οι κινητήρες χωρίς ψήκτρες (brushless motors) χρησιμοποιούνται σε εφαρμογές όπου απαιτούνται υψηλή αξιοπιστία, διάρκεια ζωής και χωρίς τον κίνδυνο σπινθήρων, λόγω της κατασκευής του κινητήρα (μόνιμοι μαγνήτες στον στροφέα). Στους συγκεκριμένους κινητήρες οι σχέσεις μεταξύ ρεύματος/ροπής και συχνότητας/ταχύτητας είναι γραμμικές. Σημαντικός παράγοντας της αυξημένης απόδοσης του κινητήρα, είναι η χρήση του ηλεκτρονικού ελεγκτή για την οδήγηση του, όπως αναλύεται παρακάτω. Επίσης, ένα χαρακτηριστικό γνώρισμα των συγκεκριμένων κινητήρων για τον διαχωρισμό μεταξύ τους είναι η σταθερά KV. Η συγκεκριμένη σταθερά μας πληροφορεί τις στροφές του κινητήρα χωρίς φορτίο ανά βολτ τροφοδοσίας. Για παράδειγμα, κινητήρας με σταθερά 1000KV και τάση 11Volt στρέφεται με $11 \times 1000 = 11.000$ rpm (στροφές ανά λεπτό). Με γνώμονα αυτήν τη σταθερά και τις στροφές που θέλουμε να επιτύχουμε, μπορούμε να επιλέξουμε τον κινητήρα που καλύπτει τις απαιτήσεις της εκάστοτε εφαρμογής.

Πλεονεκτήματα DC brushless κινητήρα έναντι DC brush κινητήρα :

1. Μεγαλύτερη διάρκεια ζωής λόγω της απουσίας του συλλέκτη.
2. Μεγαλύτερη αποδιδόμενη ροπή στο ονομαστικό φορτίο.
3. Παράγουν χαμηλό ηλεκτρικό θόρυβο.
4. Οι ονομαστικές τους στροφές μπορεί να είναι υψηλές (60000 rpm) με συνδυασμό ικανοποιητική ροπή.
5. Περισσότερη ροπή ανά μονάδα ισχύος.

Μειονεκτήματα:

1. Απαιτήση αισθητήρων και ανάδρασης για την οδήγηση τους.
2. Χαμηλή απόδοση σε χαμηλή τάση λειτουργίας.
3. Μεγαλύτερο κόστος.
4. Η αντιστροφή φοράς περιστροφής είναι πολύπλοκη (αλλαγή 2 φάσεων μεταξύ τους).

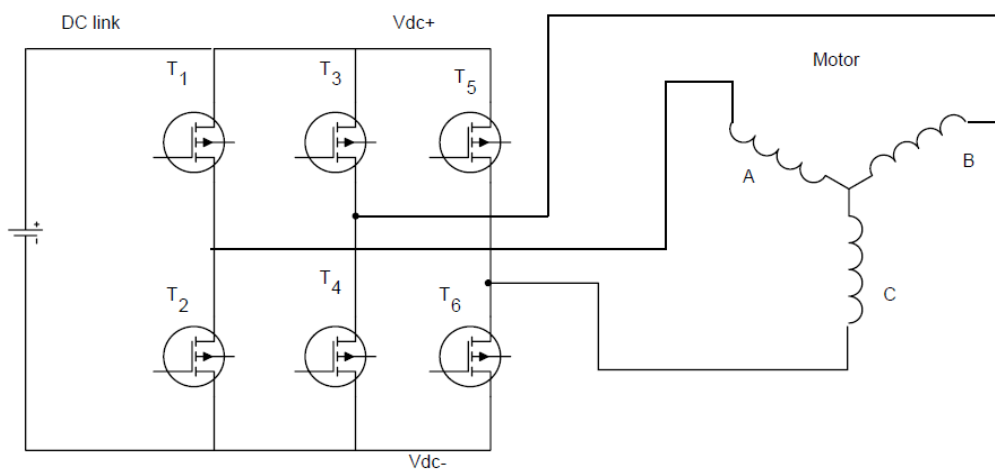
Οδήγηση Brushless κινητήρων

Η ρύθμιση των στροφών του κινητήρα γίνεται με τρανζίστορ ισχύος (Mosfet), λογικά κυκλώματα και έναν μικροελεγκτή τα οποία περιλαμβάνονται στην πλακέτα του ESC (Electronic Speed Controller). Το ESC με την σειρά του δέχεται ένα σήμα PWM συχνότητας 50Hz, του οποίου οι παλμοί ποικίλουν από 1ms-2ms, αναλόγως το επιθυμητό ποσοστό λειτουργίας σε κάθε χρονική στιγμή. Τα χαρακτηριστικά των ESC που είναι σημαντικά και τα διαφοροποιούν μεταξύ τους, είναι το ρεύμα με το οποίο τροφοδοτούν έναν κινητήρα και η τάση τροφοδοσίας τους. Τέλος, για την οδήγηση ενός brushless κινητήρα ο μικροελεγκτής που βρίσκεται στο ESC τροφοδοτεί με παλμούς PWM τα 6 Mosfet, τα οποία είναι σε συνδεσμολογία ημιγέφυρας (half-bridge) και με συγκεκριμένη χρονική σειρά ενεργοποίησης

των τρανζίστορ, όπως φαίνεται παρακάτω στον πίνακα επιτυγχάνεται η οδήγηση του κινητήρα στις επιθυμητές στροφές.

Διάστημα	Τρανζίστορ που άγουν	Είσοδος	Έξοδος
1	T1,T4,T6	A	B,C
2	T1,T3,T6	A,B	C
3	T3,T2,T6	B	A,C
4	T2,T3,T5	B,C	A
5	T2,T4,T5	C	A,B
6	T1,T4,T5	A,C	B

Πίνακας 1.Ακολουθία οδήγησης τρανζίστορ



Σχήμα 21 Συνδεσμολογία των τρανζίστορ στα αριστερά με τα τυλίγματα του κινητήρα δεξιά

Το ESC που χρησιμοποιήθηκε στην υλοποίηση της πλατφόρμας με τα τεχνικά χαρακτηριστικά του εικονίζεται στο παρακάτω σχήμα.



Συνεχόμενο ρεύμα	25Ax4
Ρεύμα κορυφής	30Ax4(>10s)
□EC□mode	Switch mode
Έξοδο□ BEC	5V/3A
Κελιά μπαταρίας	2S-4S(7.□□-14.8□)

(α)

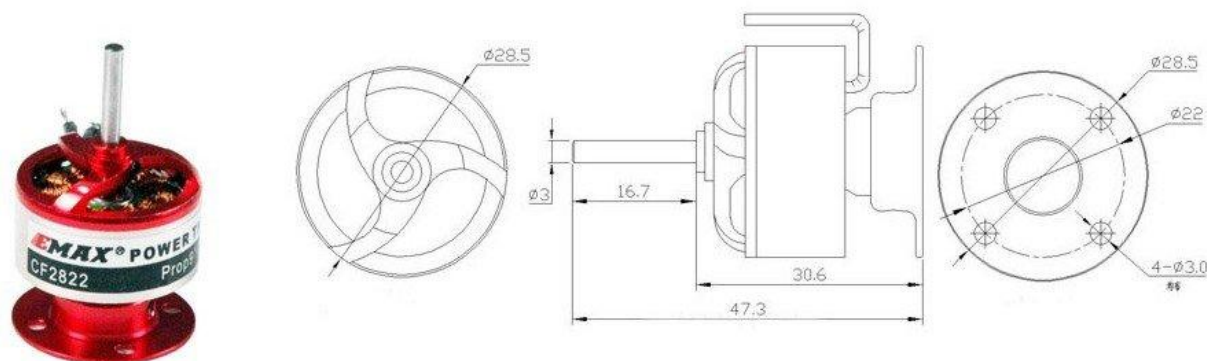
(β)

Σχήμα 22 (α) Ηλεκτρονική μονάδα ελέγχου ταχύτητας κινητήρων brushless (electronic speed controller, ESC).

(β)Τεχνικά χαρακτηριστικά της μονάδας ESC.

Βαθμονόμηση λειτουργίας DC κινητήρα BRUSHLESS

Ο κινητήρας που χρησιμοποιήθηκε στην υλοποίηση της πλατφόρμας, μαζί με τα τεχνικά του χαρακτηριστικά, παρουσιάζονται στο επόμενο σχήμα.



No. Of cells	2-3X LI-Poly	Model	Cell Count	RPM/V	Prop (APC)	RPM	MAX current (<60S)	Thrust
Stator dimensions	22x10mm	CF2822	3S	1200	10x5	7700	15A	840g 1.85lb.
Shaft diameter	3mm				10x4.7 (SLOW)	7100	16.5A	850g 1.87lb.
Weight	39g/1.38oz.	CF2812	3S	1534	7x4	12250	10.5A	580g 1.27lb.
Recommended model weight	200-600g				7x6	11000	15.5A	730g 1.61lb.
Recommended prop without gearbox	APC 7x6 APC 8x3.8 APC 9x4.5 APC10x5							

Σχήμα 23 EMAX CF2822 1200KV

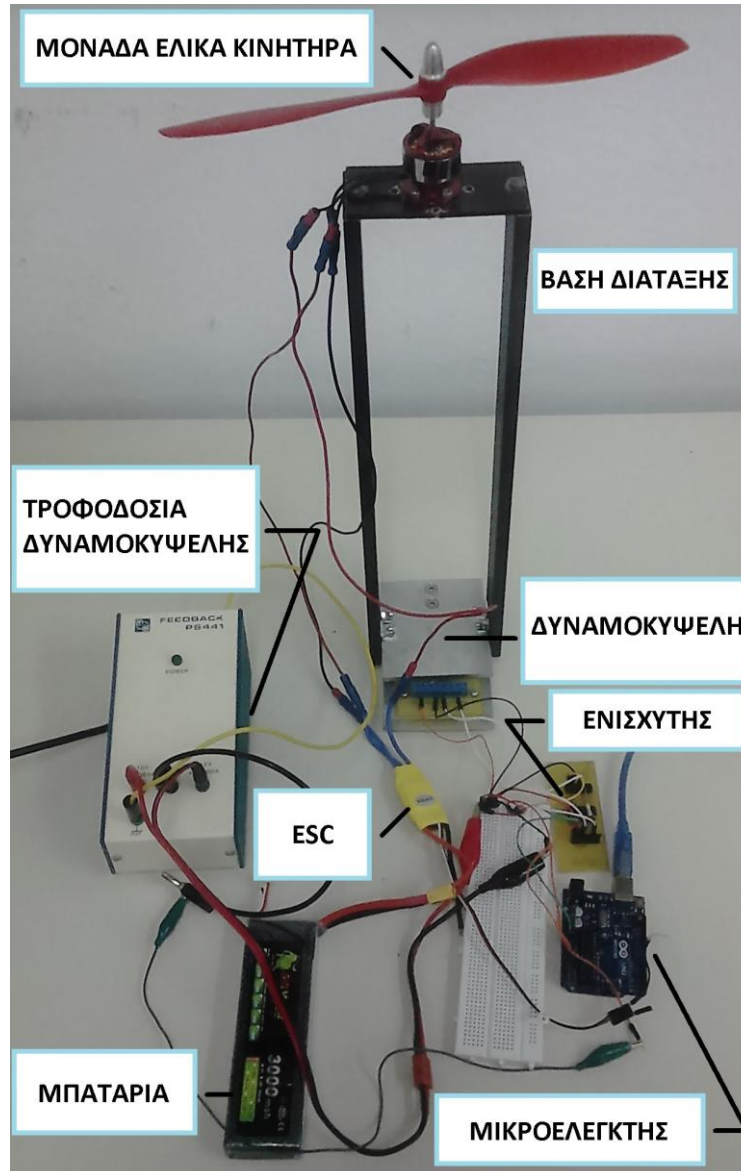
Λόγω διαφοροποιήσεων που παρουσιάζουν μεταξύ τους οι κινητήρες, είναι σημαντική η βαθμονόμησή τους για την επίτευξη ευσταθέστερου ελέγχου της πλατφόρμας. Για τον λόγο αυτό πραγματοποιήθηκαν 2 πειράματα, όπου με τη χρήση δυναμοκυψέλης (load cell) και της μονάδας έλικας/κινητήρα, επιτεύχθηκε ένα κοινό μοτίβο ελέγχου. Το πρώτο πείραμα πραγματοποιήθηκε με σκοπό τη μέτρηση της ώσης που παράγει η κάθε μονάδα ανα 10% λειτουργίας. Στη συνέχεια από τις μετρήσεις υπολογίστηκαν οι συντελεστές ενός πολυωνόμου 3^{ου} βαθμού. Το δεύτερο πείραμα πραγματοποιήθηκε για την επιβεβαίωση της βαθμονόμησης, όπου με τη χρήση του πολυωνόμου στον κώδικα του μικροελεγκτή, υπολογίζεται η τιμή του Duty cycle που απαιτείται για την εκάστοτε επιθυμητή ώση.

ΠΕΙΡΑΜΑΤΑ ΒΑΘΜΟΝΟΜΗΣΗΣ

Εξαρτήματα για την υλοποίηση των πειραμάτων:

- Μικροελεγκτής Arduino Uno
- Κινητήρες brushless CF2822 1200KV.
- Δυναμοκυψέλη (load-cell) 0-5kg.
- Τροφοδοτικό +-15V.
- Ενισχυτής τάσης δυναμοκυψέλης INA101.
- Έλικα 10x45.
- ESC 3s τάσης 11,1V 30A.
- Πλαίσιο αλουμινίου στήριξης μονάδας έλικα/κινητήρα.

Η διάταξη των πειραμάτων με τη μονάδα έλικα/κινητήρα και τη δυναμοκυψέλη (load-cell) εικονίζονται στο ακόλουθο σχήμα.



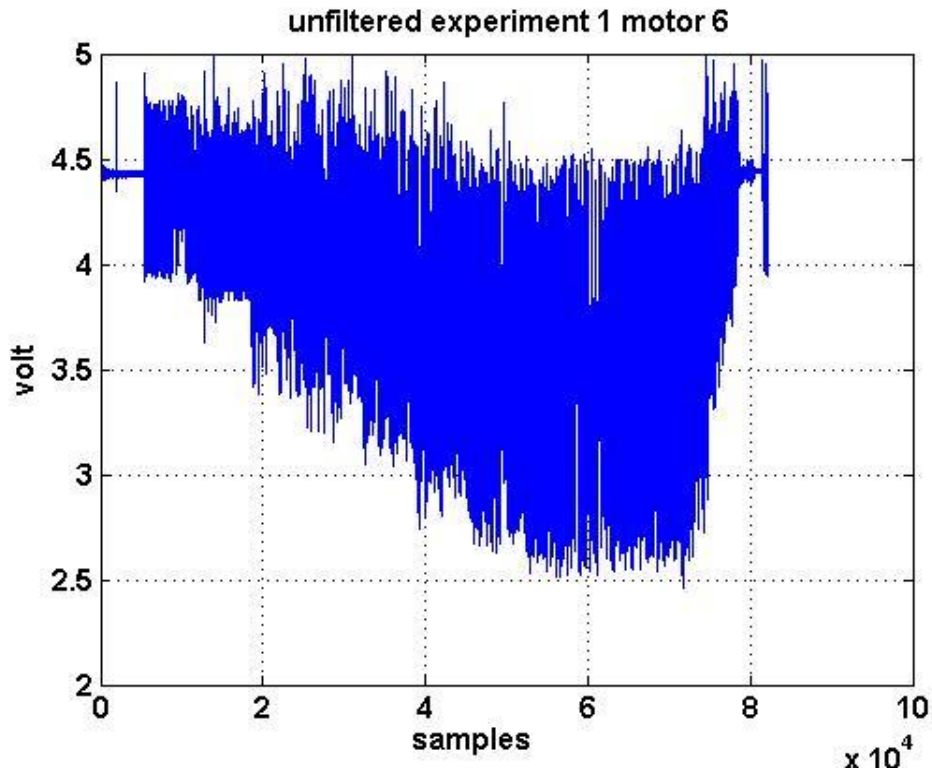
Σχήμα 24 Η διάταξη που υλοποιήθηκε για τα πειράματα

ΠΕΙΡΑΜΑ 1.

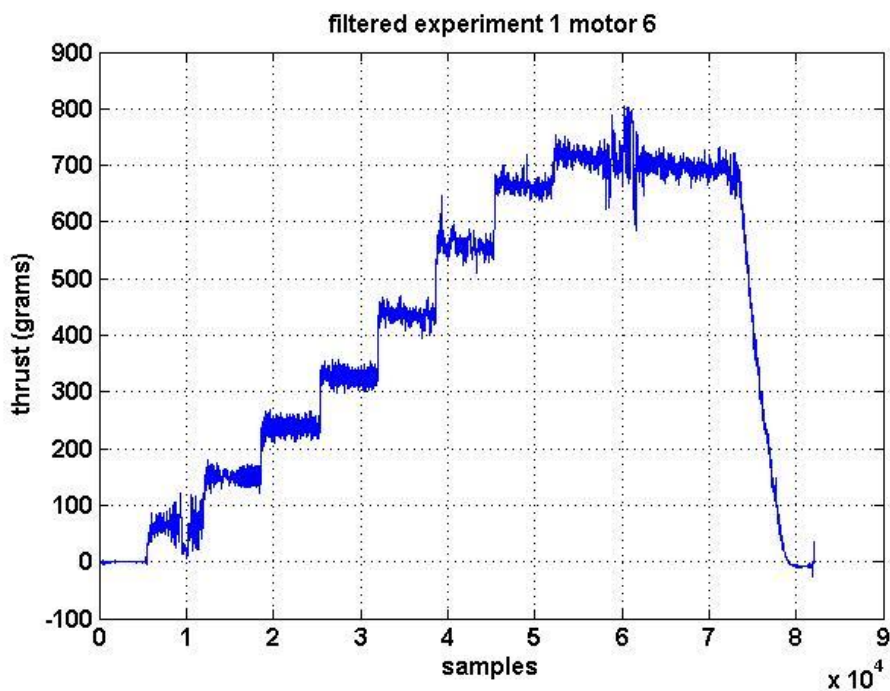
Στο συγκεκριμένο πείραμα, ο μικροελεγκτής Arduino Uno τροφοδότησε το ESC, με σήμα PWM μεταβλητό από 0-100% με βήμα 10%. Η δυναμοκυψέλη αποτελείται από μια γέφυρα αντιστάσεων. Η διαφορά τάσης που δημιουργείται κατά την διάρκεια του πειράματος, είναι πολύ μικρή για να μετρηθεί απευθείας από τον μικροελεγκτή. Με την τροφοδοσία της γέφυρας της κυψέλης με +15 βολτ, ο ενισχυτής INA101 ενίσχυσε την διαφορά τάσης της γέφυρας σε επίπεδο τάσης TTL. Έτσι ο μετατροπέας αναλογικού σε ψηφιακό σήμα του μικροελεγκτή, μπορούσε πλέον να μετρήσει τη δύναμη ώσης που δημιούργησε η μονάδα έλικα/κινητήρα και να καταγραφούν οι μετρήσεις σε αρχείο.

Στη συνέχεια με χρήση του MATLAB, οι μετρήσεις από το load-cell (σχήμα25) φιλτραρίστηκαν, μετατράπηκαν από τάση σε γραμμάρια δύναμης ώσης και

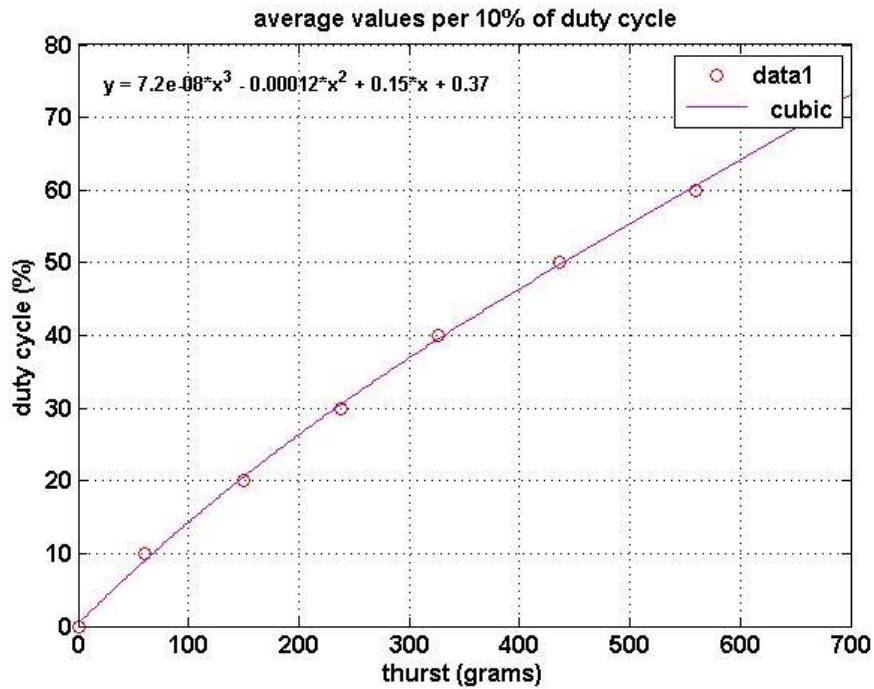
αντιστοιχήθηκαν με την εκάστοτε τιμή του PWM (σχήμα 26). Ανά στάθμη 10% υπολογίστηκε ο μέσος όρος των τιμών της εκάστοτε στάθμης, ο οποίος χρησιμοποιήθηκε για τον υπολογισμό του πολυωνόμου. Οι συντελεστές του πολυωνόμου υπολογίστηκαν με την εντολή polyfit. Η καμπύλη μέσων τιμών duty cycle σε σχέση με τη δύναμη ώσης που μετρήθηκε παρουσιάζονται στο σχήμα 27.



Σχήμα 25 Αφιλτράριστο σήμα από πείραμα 1



Σχήμα 26 Μετρήσεις ώσης σε γραμμάρια σε γράφημα από το πείραμα 1



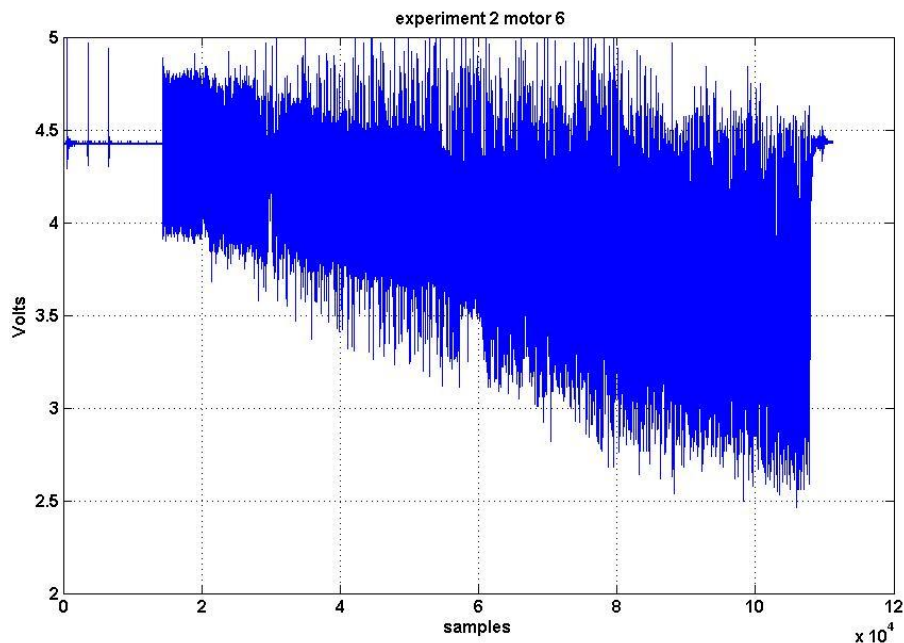
Σχήμα 27 Μέσες τιμές των γραμμαρίων ώσης από πείραμα 1

ΠΕΙΡΑΜΑ 2.

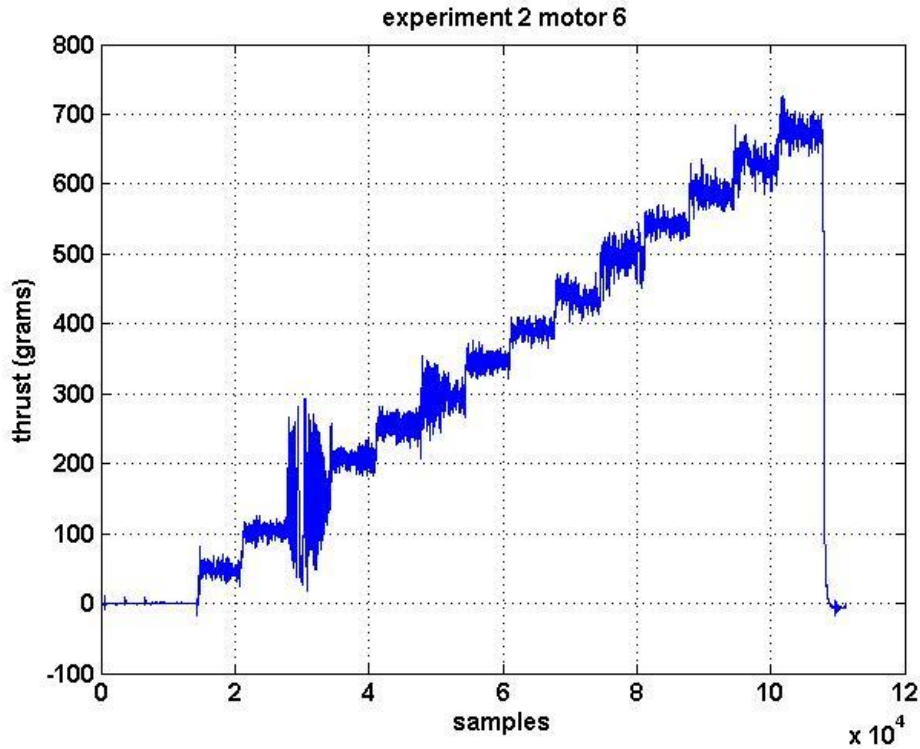
Σκοπός του δεύτερου πειράματος είναι η επιβεβαίωση της βαθμονόμησης, όπου με χρήση του πολυώνμου από το πείραμα 1 και επιθυμητές τιμές δύναμης από 0-700gr με βήμα 50gr, έγινε διασταύρωση των μετρήσεων με των επιθυμητών τιμών του πειράματος.

Τα δύο πειράματα υλοποιήθηκαν και για τους τέσσερις κινητήρες με τις μετρήσεις και τα πολυώνυμα τους να παρατίθενται στο εκάστοτε παράρτημα.

Στα ακόλουθα σχήματα παρουσιάζονται τα αποτελέσματα από το πείραμα 2.



Σχήμα 28 Αφιλτράριστες μετρήσεις από πείραμα 2



Σχήμα 29 Γράφημα επιβεβαίωσης βαθμονόμησης από πείραμα 2

Στο παραπάνω γράφημα εικονίζεται η βαθμονομημένη λειτουργία του κινητήρα. Στον ακόλουθο πίνακα παρουσιάζονται τα πολυώνυμα ανά μονάδα έλικα/κινητήρα.

Κινητήρας	χ^3	χ^2	χ^1	χ^0
κινητήρας M3	3.1057e-09	-8.4270e-05	0.1485	0.0969
κινητήρας M6	7.23e-08	-1.1694e-04	0.1503	0.3691
κινητήρας M7	3.6572e-08	-7.5659e-05	0.1416	0.0070
κινητήρας M8	1.5288e-07	-1.8153e-04	0.1585	-0.2505

Πίνακας 2 Πολυώνυμα ανά μονάδα έλικα/κινητήρα

2.4 Μπαταρία

Γενικές πληροφορίες

Οι μπαταρίες Ιόντων πολυμερών Λιθίου (Li-Po) είναι επαναφορτιζόμενες μπαταρίες και αποτελούνται από πολλά κελιά παράλληλα μεταξύ τους έτσι ώστε να αυξηθεί η ικανότητα αποφόρτισης της μπαταρίας.

Οι μπαταρίες αυτές είναι η εξέλιξη των μπαταριών Ιόντων Λιθίου και χρησιμοποιούνται ευρέως από το 1995. Η διαφορά τους με τις μπαταρίες Ιόντων Λιθίου είναι ότι ο ηλεκτρολύτης δεν βρίσκεται σε οργανική μορφή, αλλά σε στερεό σύνθετο πολυμερές και μερικά προτερήματά τους είναι:

το χαμηλότερο κόστος κατασκευής.

η ικανότητα συσκευασίας σε πολλές μορφές.
αξιοπιστία και αντοχή.

Ικανότητα αποφόρτισης σε πολύ μικρό χρονικό διάστημα της τάξης των 30 ή ακόμα και 50 φορές της χωρητικότητας των κελιών της μπαταρίας.

Το τελευταίο προτέρημα τους είναι και ένα από τα χαρακτηριστικά με βάση τα οποία επιλέγουμε μία μπαταρία Li-Po, ο επονομαζόμενος ρυθμός αποφόρτισης (25C,30C,50C,κ.α.) ο οποίος μετριέται σε amperes.

Τα χαρακτηριστικά επιλογής μιας μπαταρίας Li-Po είναι, η τάση της μπαταρίας, η χωρητικότητα, ο ρυθμός αποφόρτισης και η συνδεσμολογία των κελιών της μπαταρίας. Σημαντικό ρόλο στον χρόνο πτήσης της πλατφόρμας παίζουν το βάρος και η χωρητικότητα της μπαταρίας. Για το λόγο αυτό, στη πλατφόρμα που υλοποιήθηκε χρησιμοποιήθηκαν 3 μπαταρίες με τα παρακάτω χαρακτηριστικά:

Χωρητικότητα (mAh)	3000	5500	10000
Τάση (V)	11.1	11.1	11.1
Σύνδεση κελιών	3S1P	3S1P	3S1P
Ρυθμός αποφόρτισης	25C	35C	25C
Ρυθμός φόρτισης	2C	2C	2C
Βάρος (grams)	246	475	628

Πίνακας 3 Χαρακτηριστικά μπαταριών

Στο παράρτημα παρατίθενται τα υπόλοιπα χαρακτηριστικά των συγκεκριμένων μπαταριών (διαστάσεις, κ.τ.λ.)

Φόρτιση μπαταρίας

Ένα από τα σημαντικότερα σκέλη στην ανάπτυξη της πλατφόρμας είναι η φόρτιση της μπαταρίας, για λόγους ασφάλειας χρήσης της πλατφόρμας αλλά και για μεγαλύτερη διάρκεια ζωής της μπαταρίας.

Οι μπαταρίες Li-Po για τη φόρτιση τους απαιτούν ειδικές διατάξεις για τους εξής λόγους:

- Σε περίπτωση υπερφόρτισης υπάρχει μεγάλος κίνδυνος η μπαταρία να εκραγεί.
- Οι μπαταρίες Li-Po λόγω χρησιμοποίησης πολλών κελιών απαιτούν την εξισορρόπηση της τάσης μεταξύ των κελιών ειδικά κατά το τέλος της φόρτισης.
- Για μεγαλύτερη διάρκεια ζωής της μπαταρίας

Για την φόρτιση της μπαταρίας χρησιμοποιήθηκε ο φορτιστής IMAX B6 με τα εξής χαρακτηριστικά:

Ελεγχόμενος με μικροελεγκτή

Μεγάλης ισχύος και μεγάλης επίδοσης κύκλωμα

Επιθεώρηση της τάσης για κάθε κελί ξεχωριστά

Φόρτιση τύπων μπαταρίας LiIon/LiPo/Life από 1 έως 6 κελιών

Φόρτιση τύπων μπαταρίας NiCd/NiMH από 1 έως 15 κελιών
Φόρτιση τύπου μπαταρίας Lead acid(Pb) από 2 έως 20V
Δυνατότητα αποφόρτισης
Ρυθμός φόρτισης από 0.1A έως 2A.

2.5 Έλικες

Γενικές πληροφορίες για τις έλικες

Οι έλικες με την βοήθεια των κινητήρων, δημιουργούν την απαραίτητη ώση για την πτήση και τις κινήσεις που δύνανται να κάνει η πλατφόρμα. Η έλικα με την δύναμη που ασκεί στον αέρα, τον μετατοπίζει λόγω της αρχής δράσης/αντίδραση, ο αέρας ασκεί μια ίση και αντίθετη δύναμη στην έλικα με αποτέλεσμα να έχουμε την ώση. Η ώση διαφοροποιείται αναλόγως με τις στροφές που έχει η έλικα. Τα χαρακτηριστικά μιας έλικας είναι η διάμετρος, το γεωμετρικό βήμα, το πραγματικό βήμα, το ονομαστικό βήμα και ο αριθμός των 'λεπιδών' (φύλλα).

Στη πλατφόρμα που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής χρησιμοποιήθηκαν δίφυλλες έλικες, για τον λόγο αυτό ότι αναφερθεί έχει σχέση με τις δίφυλλες μόνο.

- Η διάμετρος της έλικας είναι η απόσταση από την μία έως την άλλη άκρη.
- Το γεωμετρικό βήμα είναι η γωνία που σχηματίζεται από την χορδή της αεροτομής και το επίπεδο περιστροφής της έλικας.
- Το πραγματικό βήμα είναι η πραγματική απόσταση που προχωρεί ανά περιστροφή η έλικα στην πτήση και η οποία είναι μικρότερη από το γεωμετρικό βήμα.
- Η διαφορά μεταξύ του γεωμετρικού και πραγματικού βήματος ονομάζεται γλίστρημα, το οποίο είναι επιθυμητό σε ποσοστό 10-30%. Το γλίστρημα είναι εκείνο που παράγει την ώση της έλικας. Εξαρτάται από το σχήμα της έλικας, την ευκαμψία του υλικού, το βάρος του μοντέλου κ.α. με αποτέλεσμα, το πραγματικό βήμα να μην είναι σταθερό και να μη μπορεί να μετρηθεί. Επίσης δεν μπορεί να μετρηθεί λόγω διαφορετικού βήματος κατά μήκος της έλικας, όπου έχουμε και διαφορετικό γλίστρημα.
- Το ονομαστικό βήμα είναι το βήμα που έχει η έλικα γύρω στο 75-80% της διαμέτρου της και σε συνδυασμό με την διάμετρο χαρακτηρίζεται η έλικα για την αγορά και την χρήση της.

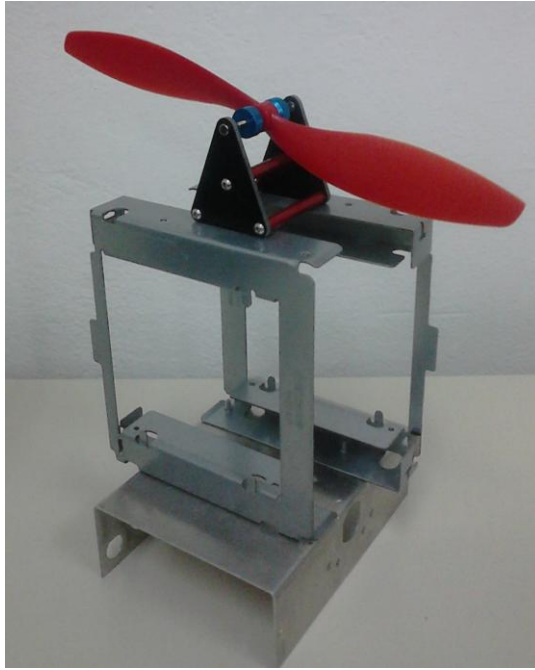
Εξισορρόπηση έλικας

Με την αγορά της, η έλικα δεν είναι ισορροπημένη. Δηλαδή υπάρχει περισσότερη μάζα σε ένα από τα δύο φύλλα με αποτέλεσμα, σε υψηλές στροφές να δημιουργούνται ανεπιθύμητες δονήσεις. Η σταθεροποίηση της έλικας είναι πολύ σημαντική, γιατί με αυτό το τρόπο μειώνονται οι δονήσεις στο σώμα της πλατφόρμας, με αποτέλεσμα:

1. Τα διάφορα μέρη της πλατφόρμας όπως οι βίδες του πλαισίου να μην ξεβιδώνονται από τις δονήσεις που δημιουργούνται.

2. Λόγω μειωμένου θορύβου των μετρήσεων του αισθητήρα, μειώνεται η κατανάλωση ισχύος για διορθωτικές κινήσεις.

Για την επίτευξη της ισορρόπησης της έλικας χρειάστηκε, η χρήση της διάταξης που εικονίζεται παρακάτω και η σταδιακή αφαίρεση μάζας από το βαρύτερο φύλλο. Με το τέλος της ισορρόπησης η έλικα είναι ικανή να τοποθετείται σε οποιαδήποτε γωνιακή θέση και να ισορροπεί, λόγω της ισοκατανομής του βάρους της.



Σχήμα 30 Η έλικα 'ζυγισμένη' στην διάταξη.

Η έλικα στην διάταξη όπου παρουσιάζεται η ανισορροπία στις λεπίδες της έλικας:



Σχήμα 31 Η έλικα στην διάταξη εξισορρόπησης

2.6 Πλαίσιο Ιπτάμενης πλατφόρμας

Πλαίσιο

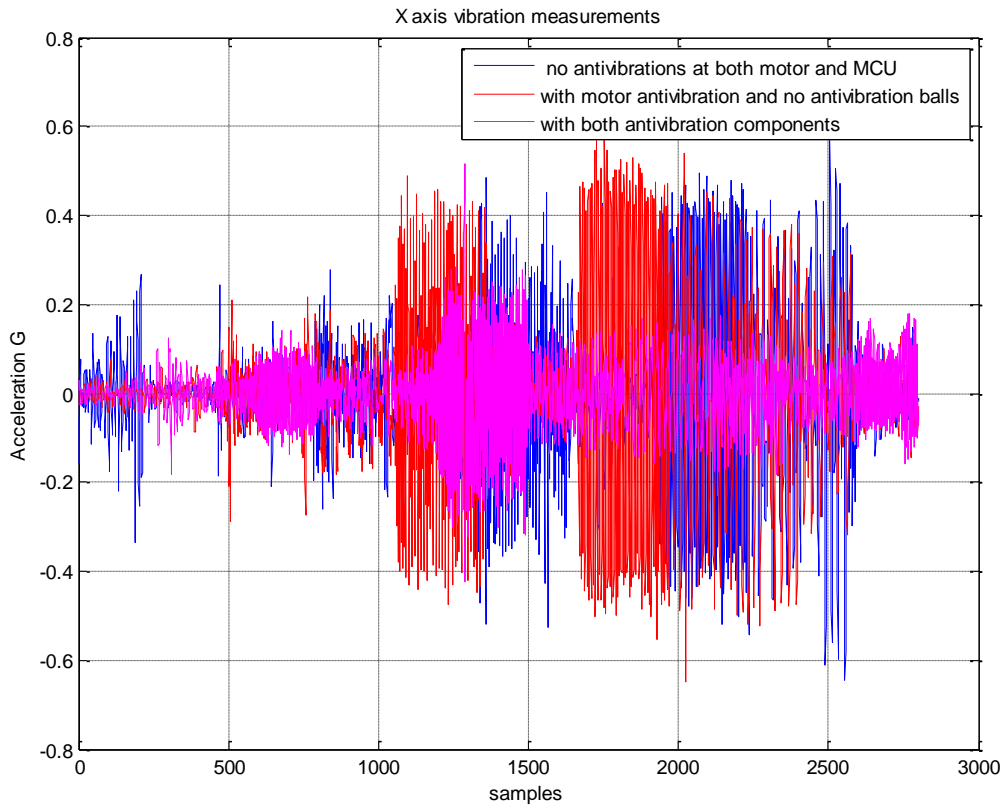
Το πλαίσιο της πλατφόρμας είναι ο σκελετός, όπου τοποθετούνται και στηρίζονται τα επιμέρους εξαρτήματα (κινητήρες, controller, esc κλπ.). Για τον λόγο αυτό πρέπει να είναι ανθεκτικός και ελαφρύς. Πάνω στον σκελετό στηρίζονται τα πόδια της πλατφόρμας, που με την σειρά τους στηρίζουν τους κινητήρες. Συνηθίζεται πάνω στα πόδια να τοποθετούνται και τα ESC. Για λόγους ελαφριάς και ισοκατανεμημένης κατασκευής αγοράστηκε έτοιμο πλαίσιο, το οποίο εικονίζεται στο ακόλουθο σχήμα:



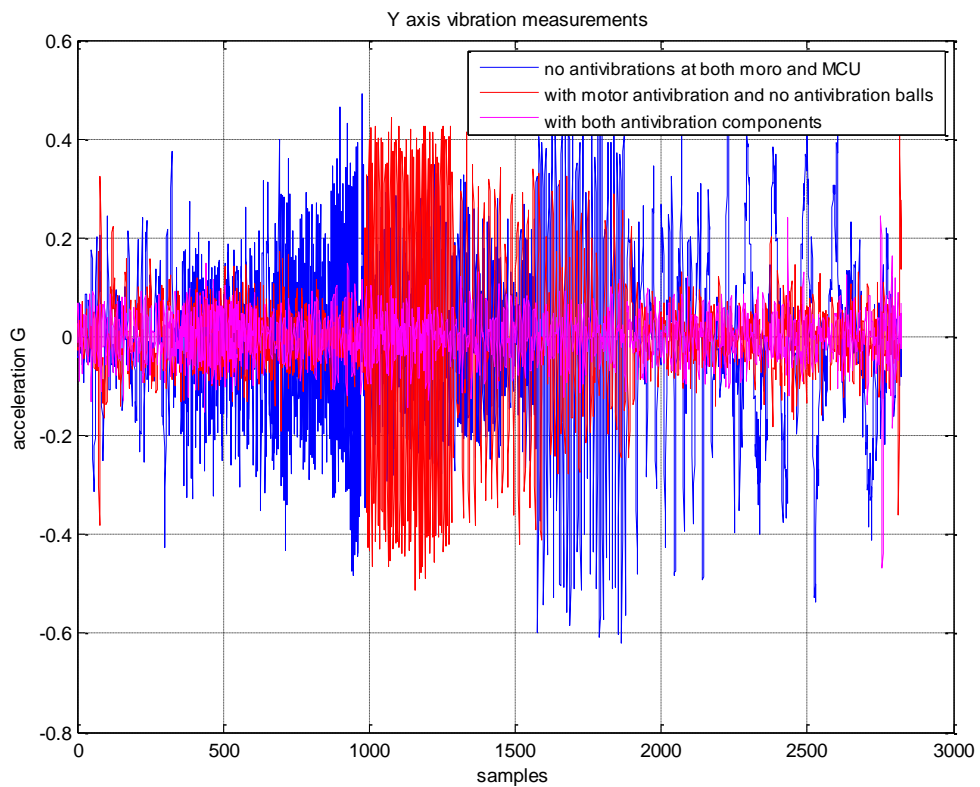
Σχήμα 32 Πλαίσιο ιπτάμενης πλατφόρμας F450 της DJI

Αντικραδασμικά

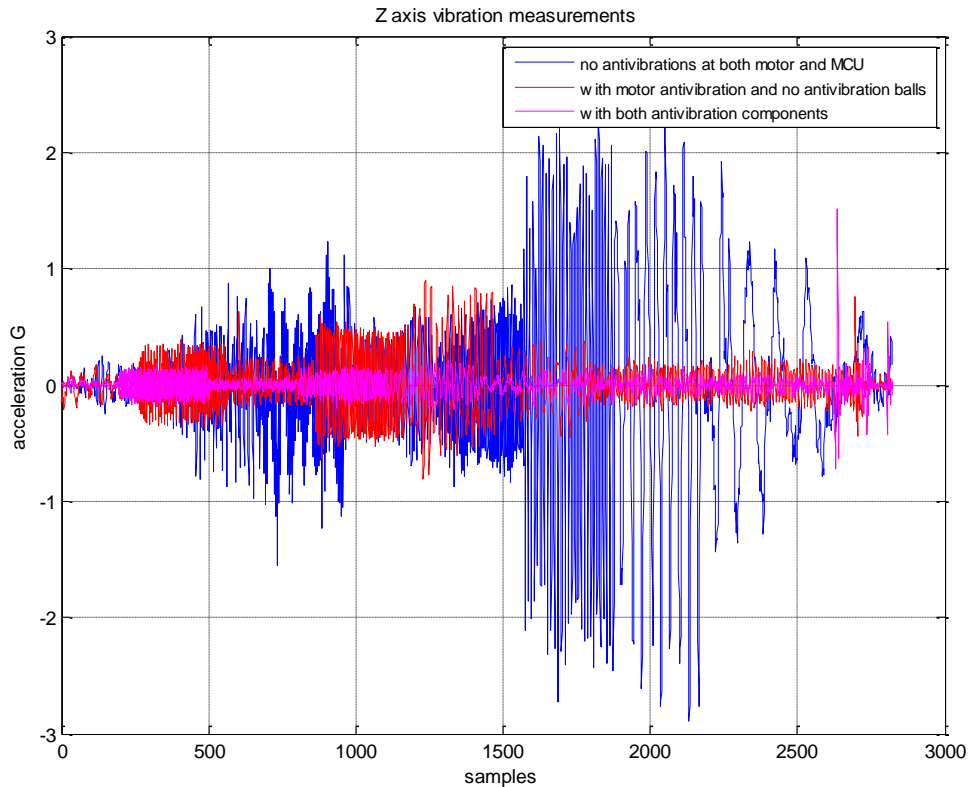
Λόγω δονήσεων που δημιουργούνται από την λειτουργία των κινητήρων, υπάρχει μεγάλος θόρυβος στις μετρήσεις του αισθητήρα, με αποτέλεσμα να κάνει περισσότερες διορθωτικές κινήσεις ο ελεγκτής πτήσης. Επίσης λόγω δονήσεων υπάρχει κίνδυνος αποσυναρμολόγησης του πλαισίου. Για να αποφευχθούν τέτοιου είδους δυσκολίες, τοποθετήθηκαν αντικραδασμικές σφαίρες στην βάση στήριξης του μικροελεγκτή και αντικραδασμικές βίδες στις βάσεις των κινητήρων. Παρακάτω παρουσιάζονται μετρήσεις του επιταχυνσιομέτρου στους τρεις άξονες, σε διάφορες περιπτώσεις, σε όλο το εύρος λειτουργίας του κινητήρα με την έλικα.



Σχήμα 33 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον X άξονα



Σχήμα 34 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον Y άξονα



Σχήμα 35 Μετρήσεις κραδασμών με την χρήση του επιταχυνσιόμετρου στον Z άξονα.

Το συμπέρασμα είναι, ότι χρειάζονται και τα δύο αντικραδασμικά συστήματα για την ομαλότερη λειτουργία του συστήματος.

Τέλος, λόγω μη συμβατότητας των κινητήρων με το πλαίσιο της πλατφόρμας, χρησιμοποιήθηκαν βάσεις από plexiglass σε συνδυασμό με τις αντικραδασμικές βίδες, όπως εικονίζεται στο ακόλουθο σχήμα:



Σχήμα 36 Ο κινητήρας με την έλικα τοποθετημένος στο πόδι του πλαισίου με το plexiglass και τα αντικραδασμικά.

Ο μικροελεγκτής στηρίχθηκε σε πλακέτα με τρύπες κατάλληλης διαμέτρου για την τοποθέτηση των αντικραδασμικών σφαιρών οι οποίες εικονίζονται στο ακόλουθο σχήμα:



Σχήμα 37 Αντικραδασμικές σφαίρες

2.7 Τηλεκατεύθυνση

Η τηλεκατεύθυνση χρησιμοποιείται για να λάβει το σύστημα της πλατφόρμας τις επιθυμητές θέσεις από τον χρήστη. Για το λόγο αυτό, το σύστημα θα πρέπει να είναι αξιόπιστο και ικανό να στέλνει τις πληροφορίες όχι μόνο σε μακρινή απόσταση αλλά και σε ύψος στην περίπτωση των υπάμενων συστημάτων.

Αποτελούμενα μέρη της τηλεκατεύθυνσης

Η τηλεκατεύθυνση αποτελείται από τον πομπό, ο οποίος στέλνει συνεχώς ένα σήμα διαμορφωμένο με εντολές, για την επιθυμητή θέση της πλατφόρμας. Ο δέκτης λαμβάνει το σήμα, το οποίο αποκωδικοποιεί και επεξεργάζεται κατάλληλα ο μικροελεγκτής. Το μεγάλο πλεονέκτημα της είναι η δυνατότητα της ταυτόχρονης λειτουργίας όλων των καναλιών, για την αποστολή και λήψη των επιθυμητών γωνιών ανά άξονα.

Το κάθε κανάλι παρέχει πληροφορίες για μία ανεξάρτητη κίνηση στην πλατφόρμα. Συγκεκριμένα χρησιμοποιήθηκαν 4 κανάλια, τα οποία είναι :

Throttle κανάλι το οποίο μεταφέρει το επιθυμητό duty cycle και των τεσσάρων κινητήρων.

Aileron κανάλι το οποίο μεταφέρει την επιθυμητή γωνία της πλατφόρμας στον Y άξονα.

Elevation κανάλι το οποίο μεταφέρει την επιθυμητή γωνία της πλατφόρμας στον X άξονα.

Rudder κανάλι το οποίο μεταφέρει την επιθυμητή γωνία της πλατφόρμας στον Z άξονα.

Ένα από τα σημαντικότερα στοιχεία της τηλεκατεύθυνσης είναι η συχνότητα επικοινωνίας μεταξύ πομπού και δέκτη. Υπάρχουν τέσσερα κριτήρια επιλογής:

1. Η συχνότητα να επιτρέπεται από τις διατάξεις του κράτους.

2. Η συχνότητα να είναι ελεύθερη από μόνιμες εξωτερικές παρεμβολές της περιοχής.
3. Να μην έχουν πολλοί άλλοι αερομοντελιστές την ίδια συχνότητα στο ίδιο μοντελοδρόμιο, για την αποφυγή μπλεξίματος μεταξύ πομπών και δεκτών.
4. Να είναι επιτρεπόμενη για υπάμενα μοντέλα σε αντιδιαστολή με τις επιτρεπόμενες μόνο για επίγεια μοντέλα.

Ο πομπός που χρησιμοποιήθηκε αποτελείται:

από δύο διαζονικά stick, με την δυνατότητα επιστροφής στο κέντρο, με την χρήση ελατηρίων στα τρία εκ των τεσσάρων καναλιών (Aileron, Elevation, Rudder)

από έναν μικροελεγκτή με μνήμη για διάφορα μοντέλα

οθόνη για καλύτερη διεπαφή με τον χρήστη

από τους επιλογείς trim με τους οποίους ο χρήστης θέτει την θέση του ουδέτερου σημείου του stick.

Διαθέτει επίσης αρκετά άλλα κουμπιά τα οποία δεν χρησιμοποιήθηκαν στη παρούσα πτυχιακή.

Η τηλεκατεύθυνση που χρησιμοποιήθηκε έχει τα εξής τεχνικά χαρακτηριστικά και εικονίζεται στο ακόλουθο σχήμα:

- Ολικό εύρος συχνότητας στα 2.4Ghz με διαμόρφωση DSM2 και 6 κανάλια
- Μνήμη δέκα μοντέλων
- Ταίριασμα μοντέλου
- Συγχρονισμός Servo
- LCD οθόνη
- Ενσωματωμένος χρονομετρητής
- Κόψιμο Throttle
- Λειτουργία εκπαίδευσης
- Αντιγραφή μοντέλου
- Μεταβαλλόμενο μήκος stick



Σχήμα 38 Πομπός συστήματος τηλεκατεύθυνσης Spektrum DX6i

Ο δέκτης της τηλεκατεύθυνσης, ο οποίος εικονίζεται στο ακόλουθο σχήμα, διασυνδέθηκε με τη πλακέτα επέκτασης του μικροελεγκτή με σκοπό, τη λήψη των σημάτων των τεσσάρων καναλιών.



Σχήμα 39 Δέκτης του συστήματος της τηλεκατεύθυνσης

2.8 Μικροελεγκτής

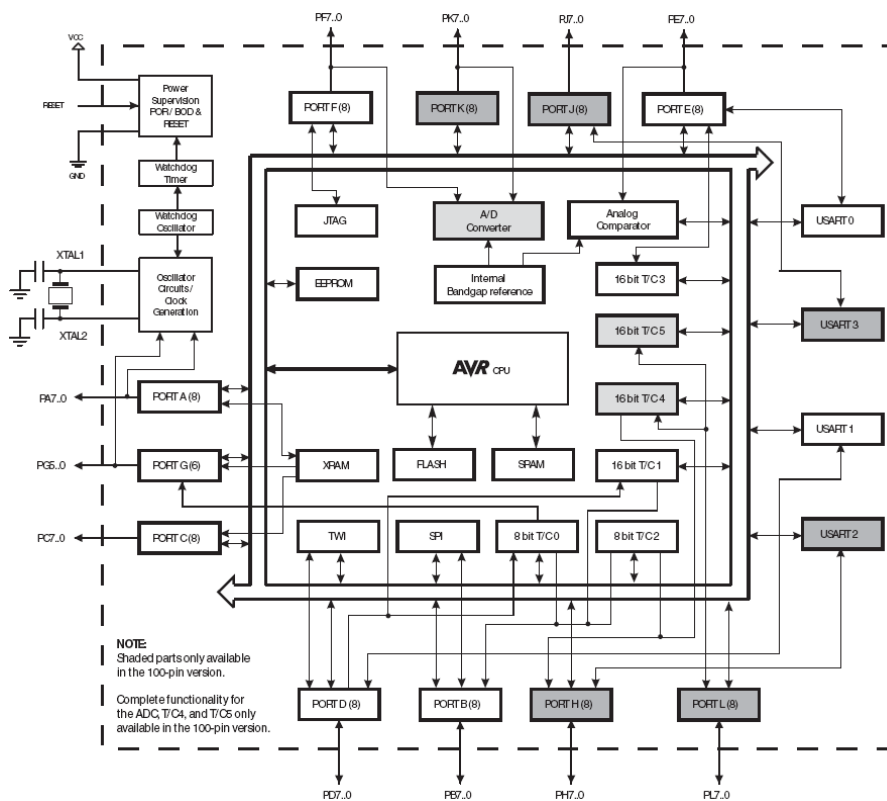
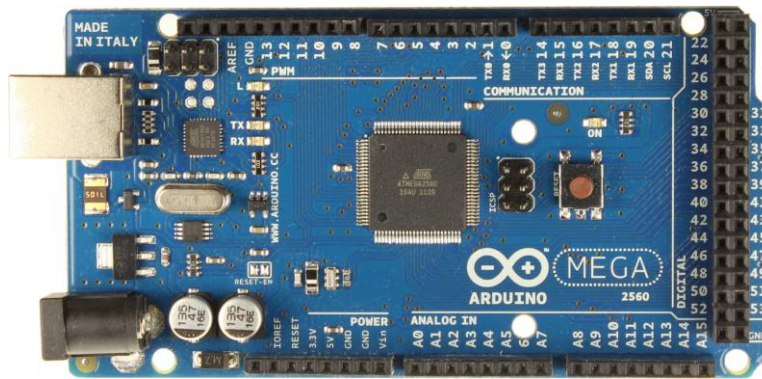
Η μικροϋπολογιστική πλατφόρμα που επιλέχθηκε για την υλοποίηση του ελεγκτή πτήσης είναι το αναπτυξιακό σύστημα Arduino Mega2560, το οποίο χρησιμοποιεί τον 8-bit μικροελεγκτή ATmega2560 της εταιρίας Atmel (Σχήμα 39).

Τα κύρια χαρακτηριστικά του αναπτυξιακού αυτού είναι:

Μικροελεγκτής ATmega2560 , με 8-bit αρχιτεκτονική RISC

Συχνότητα χρονισμού λειτουργίας 16MHz

248k μνήμη FLASH για τον κώδικα της εφαρμογής
 8k μνήμη SRAM
 4k μνήμη EEPROM
 4 χρονιστές 16-bit, 2 χρονιστές 8-bit
 14 έξοδοι hardware PWM.
 2 μονάδες 10-bit ADC, 16 αναλογικές εισοδοί
 4 μονάδες σειριακής διασύνδεσης USART
 Υποστήριξη πρωτοκόλλων SPI, I2C
 Τάση λειτουργίας 5V
 Εύρος τροφοδοσίας 7-12V
 Προγραμματισμός μέσω USB/bootloader



Σχήμα 40 Το αναπτυξιακό Arduino Mega 2560, και το λειτουργικό διάγραμμα του μικροελεγκτή

Βασικά κριτήρια της επιλογής της εν λόγω πλατφόρμας ήταν:

- Η δυνατότητα ανεξάρτητης παραγωγής μεγάλου αριθμού σημάτων PWM (μέχρι 14).
- Λήψη μέχρι 6 σημάτων μέσω εξωτερικών interrupt για τις εντολές από τον δέκτη της τηλεκατεύθυνσης, μέσω των καναλιών throttle, aileron, elevation, rudder.
- Διασύνδεση με πρωτόκολο I2C με διάφορα αισθητήρια ταυτόχρονα (γυροσκόπια, επιταχυνσιόμετρα, μαγνητόμετρα).
- Λήψη μέχρι 16 αναλογικών σημάτων για τη χρήση αισθητηρίων μέτρησης τάσης της μπαταρίας, αισθητηρίου ρεύματος, για την μέτρηση της κατανάλωσης και αναλογικού σήματος από αισθητήρια.

Επιπλέον πλεονέκτημα αποτελεί η ευκολία ανάπτυξης εφαρμογών με την εν λόγω πλατφόρμα, χάρη στο ισχυρό και εύχρηστο περιβάλλον προγραμματισμού που παρέχεται. Ειδικότερα, η παραγωγή των 4 σημάτων PWM οδήγησης των ESC και κατ' επέκταση των κινητήρων επιτυγχάνεται μέσω των 16-bit χρονιστών Timer-1,Timer-4 του μικροελεγκτή, καθένας εκ των οποίων μπορεί να παράγει τρία σήματα PWM, με κοινή συχνότητα και ανεξάρτητη ρύθμιση του duty cycle, η οποία επιτυγχάνεται καθορίζοντας κατάλληλα την τιμή των καταχωρητών σύγκρισης. Η αντιστοίχιση των σημάτων PWM με τους κινητήρες είναι η εξής:

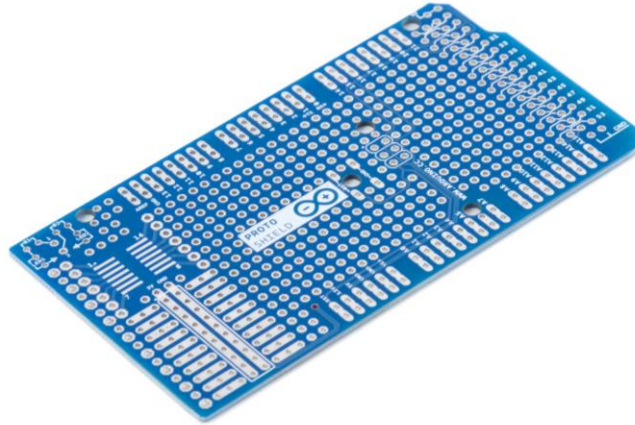
Κινητήρας	Χρονιστής	Καταχωρητής ρύθμισης duty cycle	Έξοδος στην πλακέτα
1	Timer1	OCR1A	Pin11
2	Timer1	OCR1B	Pin12
3	Timer1	OCR1C	Pin13
4	Timer4	OCR4A	Pin6

Πίνακας 4 Αντιστοίχιση σημάτων PWM με κινητήρες

Η διασύνδεση του μικροελεγκτή με τα αισθητήρια πραγματοποιήθηκε με το πρωτόκολο I2C, το οποίο χρησιμοποιεί τα pins του μικροελεγκτή 19,20.

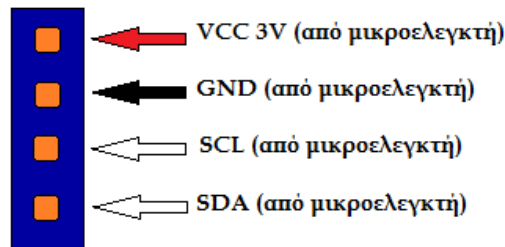
2.9 Πλακέτα επέκτασης μικροελεγκτή

Η πλακέτα επέκτασης του μικροελεγκτή που αναπτύχθηκε, παρέχει στον μικροελεγκτή την δυνατότητα προσαρμογής και διασύνδεσης των σημάτων εισόδου/εξόδου και ως βάση για τα αισθητήρια θέσης (γυροσκόπια,επιταχυνσιόμετρα). Η πλακέτα εικονίζεται στο ακόλουθο σχήμα:



Σχήμα 41 Πλακέτα επέκτασης του μικροελεγκτή

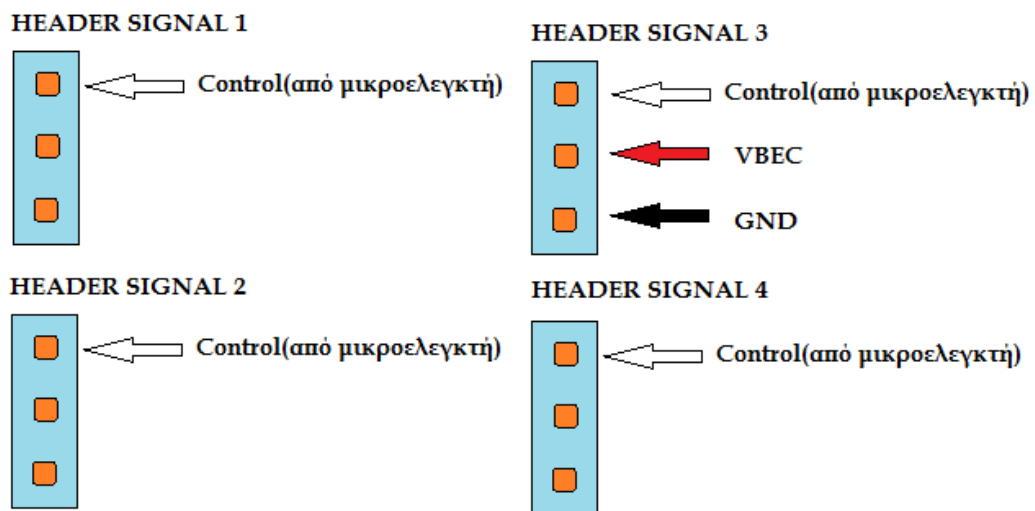
Στην μέση της πλακέτας επέκτασης τοποθετήθηκε ο αισθητήρας θέσης MPU-6050, και η διασύνδεση του με τον μικροελεγκτή εικονίζεται στο επόμενο σχήμα.



Σχήμα 42 Διασύνδεση MPU-6050 με μικροελεγκτή

Ο αισθητήρας τροφοδοτείται με 3V μέσω του πρώτου pin και του δεύτερου pin GND. Για την επικοινωνία του αισθητήρα με τον μικροελεγκτή χρησιμοποιείται το πρωτόκολλο I2C και απαιτούνται τα pin SDA (Serial DATA) SCL (Serial CLOCK) όπως εικονίζεται παραπάνω.

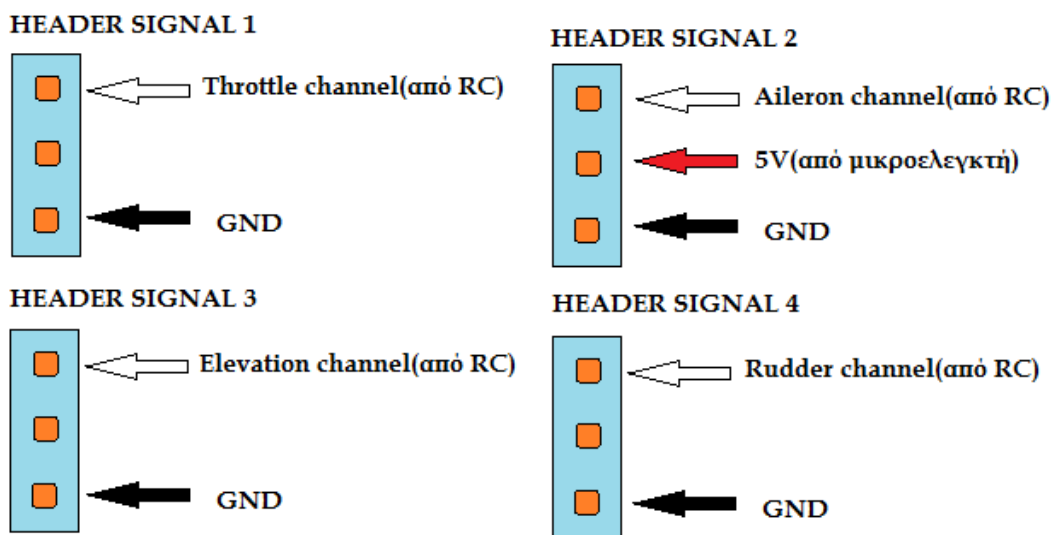
Η διασύνδεση με τον μικροελεγκτή με κάθε είσοδο του ESC, γίνεται μέσω 6 σημάτων που παρουσιάζονται στο ακόλουθο σχήμα και τα οποία παρέχονται στην πλακέτα επέκτασης μέσω 4 headers τριών ακίδων.



Σχήμα 43 Η διάταξη των 3 σημάτων ανά header για την διασύνδεση του ESC με τον μικροελεγκτή

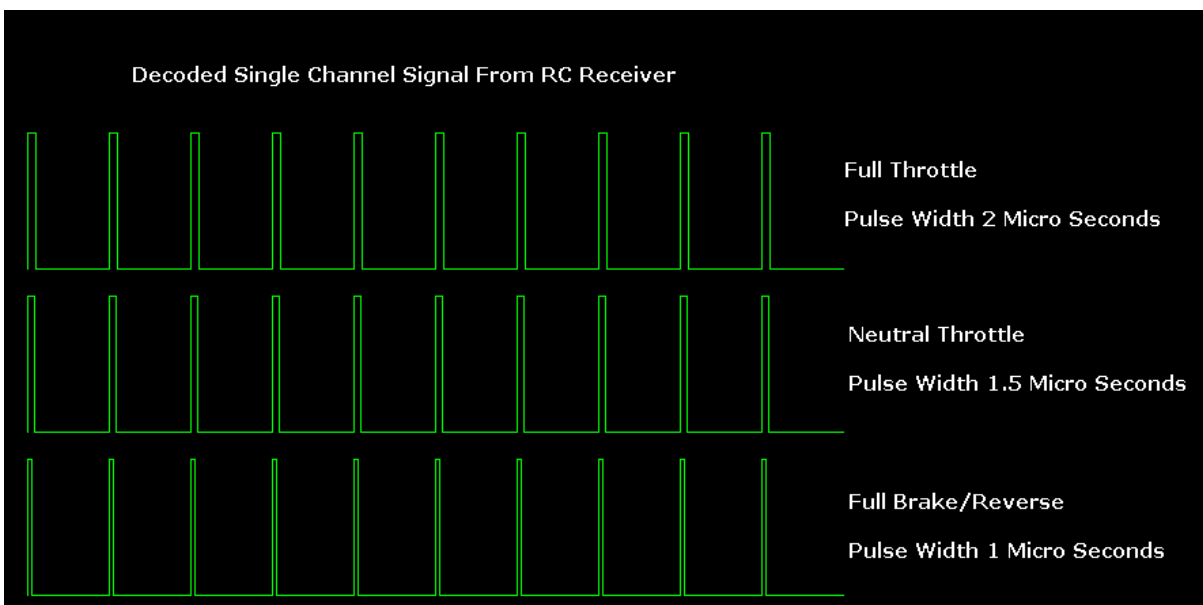
Η τροφοδοσία του μικροελεγκτή παρέχεται με την 5V τάση από το header 3 του ESC, σε συνδυασμό με το GND. Το control είναι το PWM σήμα, που δημιουργείται από τον μικροελεγκτή για τον καθορισμό της ταχύτητας του κινητήρα.

Η διασύνδεση του μικροελεγκτή με τον δέκτη της τηλεκατεύθυνσης, πραγματοποιείται μέσω 9 σημάτων που παρουσιάζονται στο ακόλουθο σχήμα και τα οποία παρέχονται στην πλακέτα επέκτασης μέσω 4 headers τριών ακίδων.



Σχήμα 44 Η διάταξη των 3 σημάτων ανά header για την διασύνδεση του δέκτη της τηλεκατεύθυνσης με τον μικροελεγκτή

Η έξοδος του δέκτη της τηλεκατεύθυνσης όπως εικονίζεται στο ακόλουθο σχήμα για ένα κανάλι, περιέχει τις πληροφορίες κατεύθυνσης για την πλατφόρμα, που αποστέλλονται από τον πιλότο και μέσω των interrupt ο μικροελεγκτής μετράει το πλάτος του σήματος.



Σχήμα 45 Σήμα στο μέγιστο, στην μέση και στο ελάχιστο από τον δέκτη της τηλεκατεύθυνσης

Για την μέτρηση των σημάτων χρησιμοποιούνται 4 εξωτερικά interrupts, ένα για κάθε κανάλι της τηλεκατεύθυνσης, τα οποία είναι στην κατάσταση ανίχνευσης αλλαγής θέσης. Αναλυτικότερα, όταν πραγματοποιείται αλλαγή στην τάση, ανεξάρτητα από το αν είναι από 5βολτ σε 0βολτ και το αντίστροφο, τότε ενεργοποιείται η συνάρτηση του interrupt. Μέσα στην συνάρτηση με την χρήση σημαιών και της συνάρτησης micros(), η οποία μετράει τον χρόνο που τρέχει το πρόγραμμα, ο μικροελεγκτής μετράει το duty cycle του σήματος. Το σήμα στη συνέχεια μεταφράζεται, σε τιμή επιθυμητής γωνιακής θέσης ή επιθυμητής δύναμης αναλόγως με το κανάλι. Στο κανάλι Throttle οι τιμές του χρόνου μεταφράζονται, σε τιμές δύναμης ώσης, με όρια από 0 έως 700. Στα κανάλια Aileron και Elevation οι τιμές μεταφράζονται σε γωνιακή θέση από -10° έως 10°, και στο κανάλι Rudder οι τιμές μεταφράζονται από -180° έως 180°.

2.10 MPU-6050 6DOF (Γυροσκόπιο, Επιταχυνσιόμετρο)

Γυροσκόπιο

Το γυροσκόπιο είναι μια συσκευή για την μέτρηση ή για την διατήρηση του προσανατολισμού, η οποία βασίζεται στις αρχές διατήρησης της στροφορμής. Το μηχανικό γυροσκόπιο είναι ένας στρεφόμενος δίσκος, του οποίου ο άξονας είναι ελεύθερος να πάρει οποιοδήποτε προσανατολισμό. Η μεγάλη χρησιμότητα του είναι ότι μπορεί να διατηρεί τον προσανατολισμό του σχεδόν σταθερό, ανεξάρτητα από την κίνηση της πλατφόρμας στην οποία στερεώνεται. Τα γυροσκόπια παρουσιάζουν μια σειρά από συμπεριφορές, όπως η κλόνιση και η μετάπτωση. Τα τελευταία χρόνια έχουν κατασκευαστεί διάφορων ειδών γυροσκόπια μερικά από αυτά είναι:

Το FOG (fiber optic gyroscope) γυροσκόπιο χρησιμοποιεί τυλιγμένη οπτική ίνα και την διαστρέβλωση του φωτός για να ανιχνεύσει μηχανική περιστροφή.

Το HRG (hemispherical resonator gyroscope) γυροσκόπιο αλλιώς επικαλούμενο γυροσκόπιο μανιτάρι, χρησιμοποιεί ένα λεπτό στέρας κατάστασης ημισφαιρικό κέλυφος πάνω σε παχύ στέλεχος. Το κέλυφος διεγείρεται σε συντονισμό από ηλεκτροστατικές δυνάμεις, προκαλούμενες από ηλεκτρόδια, τα οποία αγγίζουν σε διαφορετικές δομές λιωμένου χαλαζία, οι οποίες περιτριγυρίζουν το κέλυφος. Η μέτρηση λαμβάνεται από την αδρανειακή ιδιότητα των στάσιμων κυμάτων κάμψης.

Το RLG (ring laser gyroscope) γυροσκόπιο χρησιμοποιεί την κίνηση μιας πηγής φωτός και ενός ανιχνευτή χρώματος τοποθετημένα μαζί, με αποτέλεσμα από την συνεχή κίνηση των κυμάτων φωτός από μια κινούμενη πλατφόρμα, να υπάρξει διαφορά φάσης της ακτίνας όμοιας με την αλλαγή χρώματος, γνωστή ως αλλαγή προς το κόκκινο.

Το MEMS γυροσκόπιο παίρνει την ιδέα του Foucault pendulum και χρησιμοποιεί ένα δονούμενο στοιχείο γνωστό ως MEMS (μικροηλεκτρομηχανικό σύστημα).

Στην πλατφόρμα που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής χρησιμοποιήθηκε MEMS γυροσκόπιο 3 αξόνων.

Στα ακόλουθα σχήματα παρουσιάζονται διάφορα είδη γυροσκοπίων:



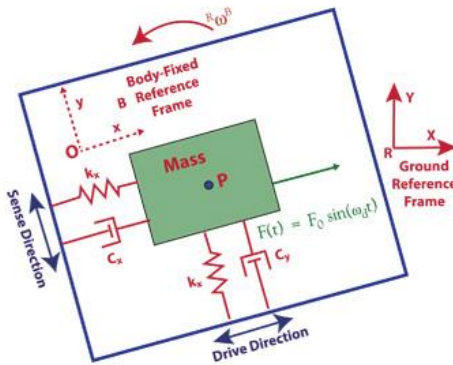
(α)



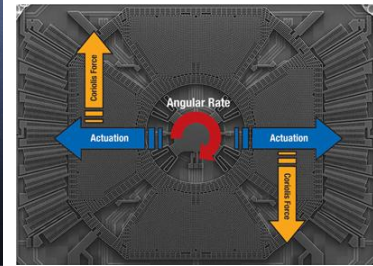
(β)



(γ)



(δ)



(ε)

Σχήμα 46 (α) γυροσκόπιο RLG (β) γυροσκόπιο HRG (γ) γυροσκόπιο FOG (δ) διάγραμμα γυροσκόπιου MEMS (ε) γυροσκόπιο 1 άξονα (MEMS)

Επιταχυνσιόμετρο

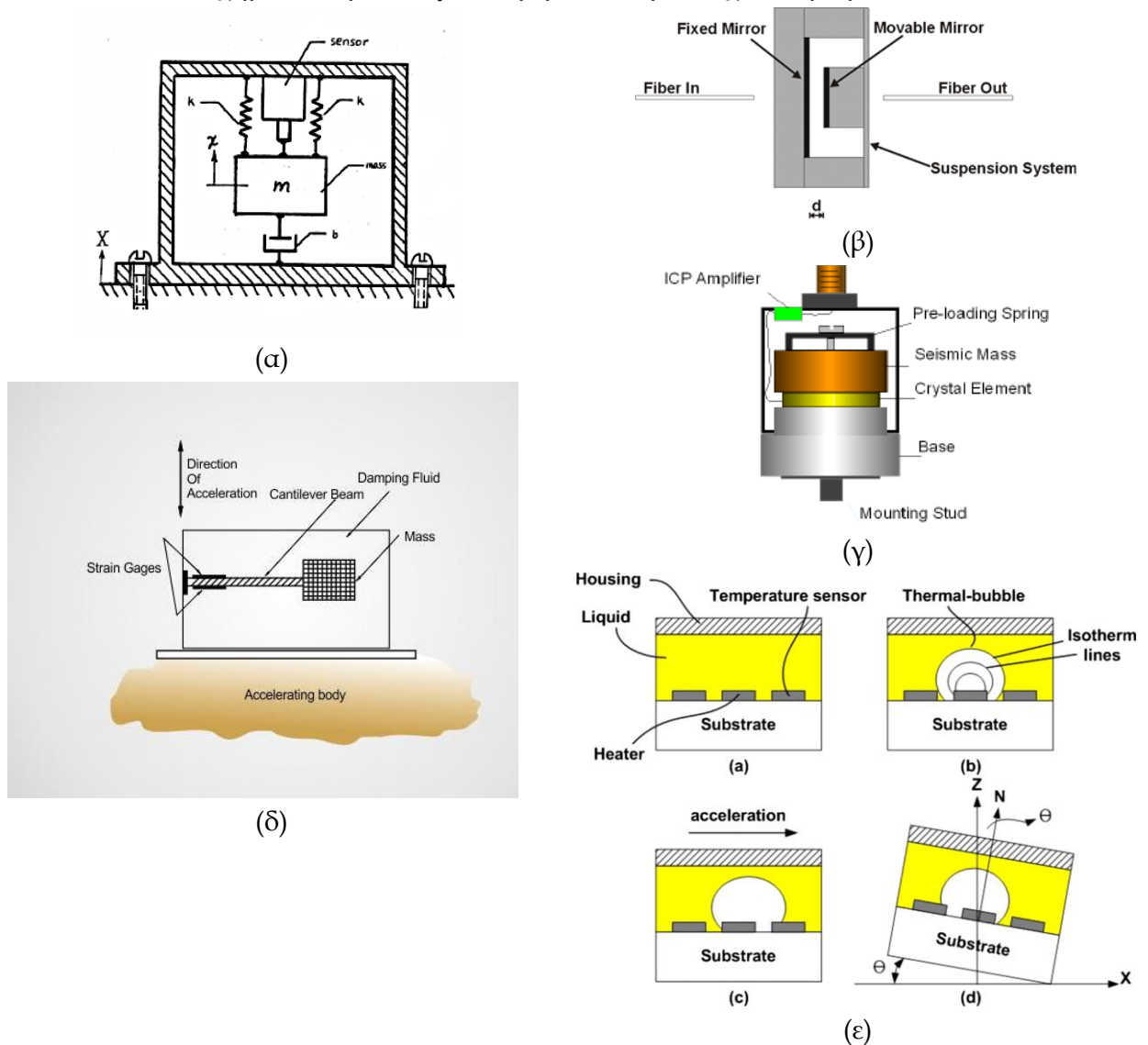
Το επιταχυνσιόμετρο είναι μια συσκευή που έχει την ικανότητα να μετρά δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως η επιτάχυνση της βαρύτητας ή δυνάμεις που προέρχονται από αλλαγές στην ταχύτητα ή στην διεύθυνση της κίνησης του αισθητηρίου (επιταχύνσεις, επιβραδύνσεις, στροφές).

Χρησιμοποιείται κυρίως σε αδρανειακά συστήματα πλοήγησης, που αποκαλούνται Inertial Navigation System (INS), τα οποία συνδυάζοντας την χρήση γυροσκοπίων και ηλεκτρονικών κυκλωμάτων, μπορούν να υπολογίσουν την θέση ενός οχήματος στον χώρο, καθώς και την ταχύτητα του με πολύ μεγάλη ακρίβεια, καθιστώντας τα απολύτως απαραίτητα εξαρτήματα, στην λειτουργία του αυτόματου πιλότου για την πλοήγηση αεροπλάνων, πλοίων και υποβρυχίων. Άλλες χρήσεις του επιταχυνσιόμετρου είναι στους αερόσακους των αυτοκινήτων και σε όποια εφαρμογή απαιτείται η μέτρηση της επιτάχυνσης. Ένα επιταχυνσιόμετρο μπορεί να κατασκευαστεί αξιοποιώντας το πιεζοηλεκτρικό φαινόμενο που δημιουργείται, όταν μάζα ανάλογη της επιτάχυνσης πιέζει έναν πιεζοκρύσταλλο με αποτέλεσμα, να δημιουργείται τάση λόγω του πιεζοηλεκτρικού φαινομένου. Διαφορετικός τύπος κατασκευής επιταχυνσιόμετρου είναι με την αλλαγή της χωρητικότητας ενός πυκνωτή, μέσω ανάρτησης μάζας και η τιμή του να μεταβάλλεται με την κίνηση της μάζας. Αντίστοιχοι τρόποι με τον προηγούμενο είναι:

- με χρήση αντίστασης για τη μέτρηση της πίεσης που δέχεται
- η χρήση φουσαλίδας ζεστού αέρα

η χρήση των ιδιοτήτων του φωτός.

Στα ακόλουθα σχήματα παρουσιάζονται μερικά είδη επιταχυνσιόμετρων:



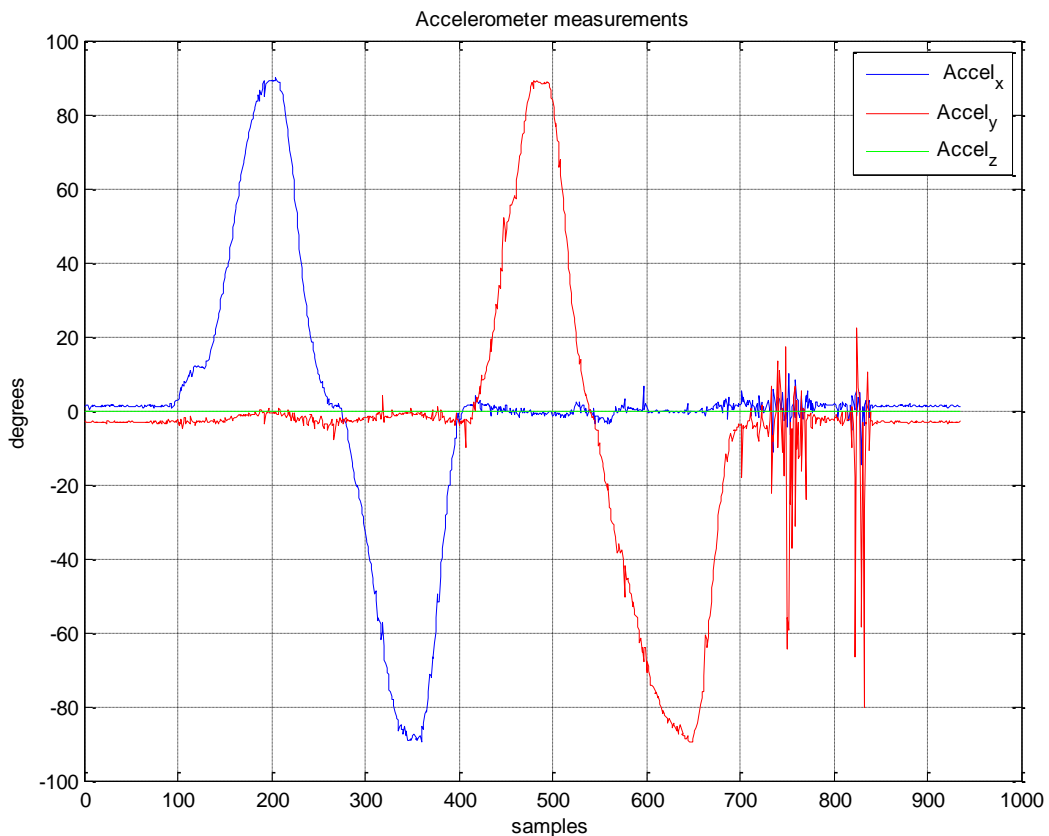
Σχήμα 47 (α) Χωρητικό Επιταχυνσιόμετρο (β) Οπτικό Επιταχυνσιόμετρο (γ) Πιεζοηλεκτρικό Επιταχυνσιόμετρο (δ) Επιταχυνσιόμετρο Αντίστασης (ε) Επιταχυνσιόμετρο Φυσαλίδας Αέρα

Αδρανειακή Μονάδα Μέτρησης

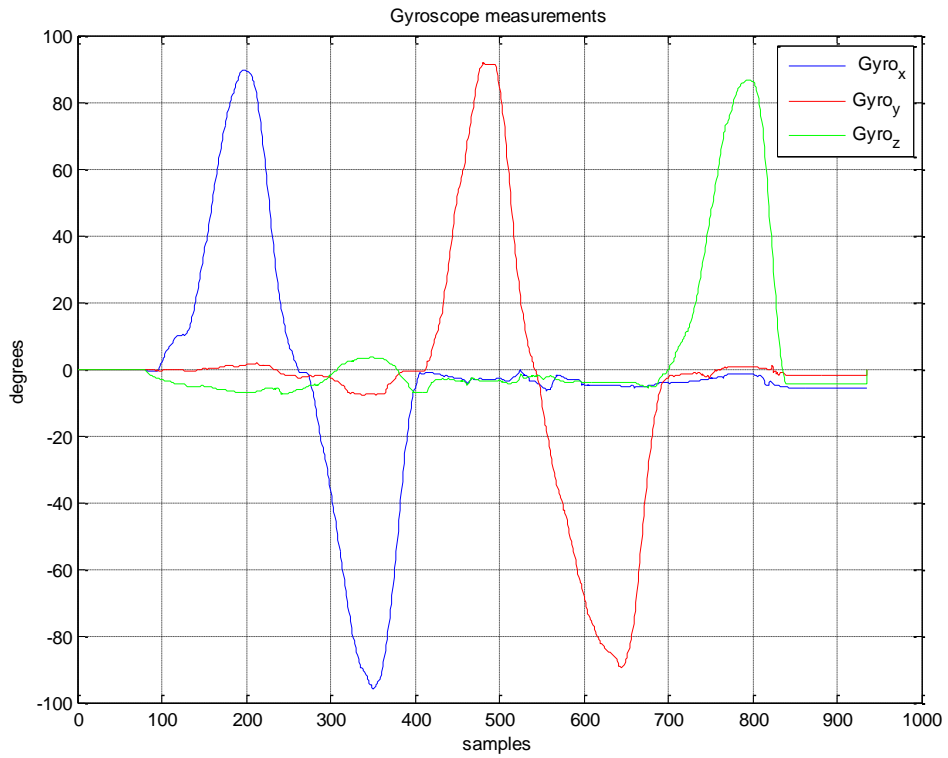
Η αδρανειακή μονάδα μέτρησης ή αλλιώς Inertial Measurement Unit (IMU) είναι, ο συνδυασμός των γυροσκοπιών τα οποία μετράνε την γωνιακή ταχύτητα, πόσο γρήγορα ένα αντικείμενο περιστρέφεται γύρω από έναν άξονα, με τα επιταχυνσιόμετρα τα οποία μετρούν δυνάμεις ανά άξονα όμως δεν μας δίνουν προσανατολισμό. Συνδυάζοντας αυτές τις δύο συσκευές επιτυγχάνεται ο ακριβείς υπολογισμός της θέσης στο χώρο του οχήματος, ή κάποιου βραχίονα ή της υπάμενης πλατφόρμας στην συγκεκριμένη περίπτωση. Για την υλοποίηση υπάρχουν τα εξής προβλήματα:

- Η γωνιακή θέση από το γυροσκόπιο παρουσιάζει μετάπτωση, λόγω υπολογισμών της θέσης από την μέτρηση της γωνιακής ταχύτητας και όχι απευθείας μέτρησης της θέσης, καθιστώντας το αξιόπιστο μόνο για μικρό χρονικό διάστημα
- Η μετρήσεις του επιταχυνσιόμετρου είναι πολύ θορυβώδεις, με αποτέλεσμα να χρησιμοποιούμε μόνο τις μακροπρόθεσμες μετρήσεις του, μέσω ενός χαμηλοπερατού φίλτρου.

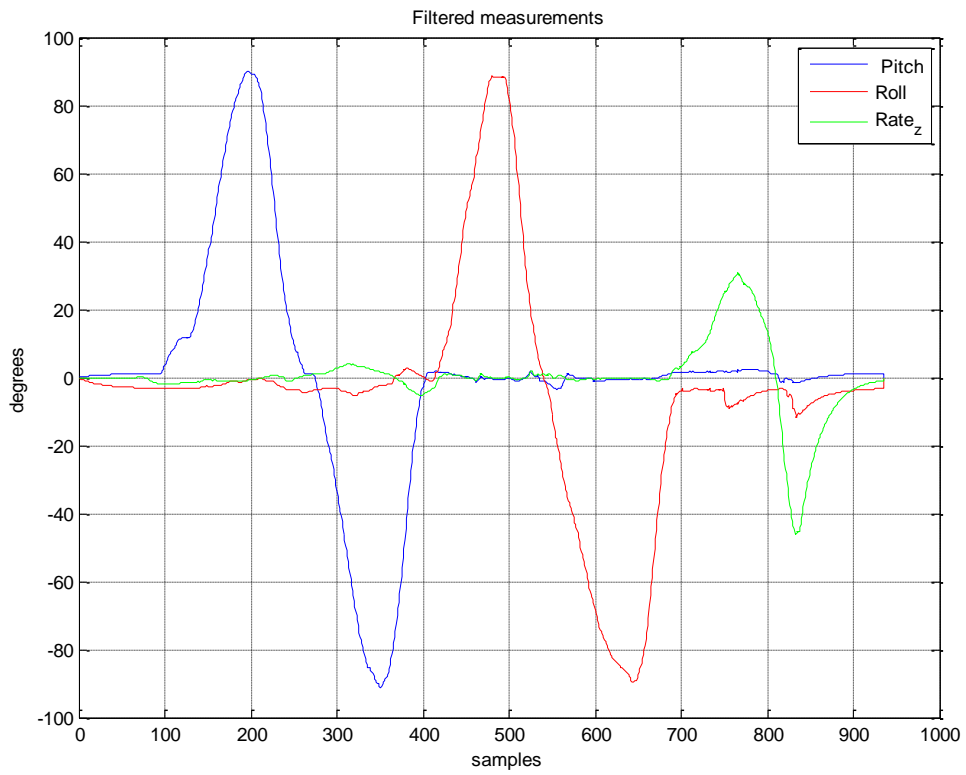
Η λύση λοιπόν βρίσκεται στον συνδυασμό των δύο μετρήσεων με την χρήση φίλτρων, όπως το complimentary filter και το [A2]extended Kalman filter. Στην παρούσα πτυχιακή χρησιμοποιήθηκε το complimentary filter στο οποίο εισάγονται οι μετρήσεις από τις δύο συσκευές με αποτέλεσμα, οι υπολογισμοί της γωνιακής θέσης του γυροσκοπίου, με την προσθήκη των μετρήσεων του επιταχυνσιόμετρου μεταβάλλονται σε αξιόπιστες, εκτός από βραχυπρόθεσμα και μακροπρόθεσμα. Σημειώνεται ότι, στον Z άξονα λόγω ύπαρξης της επιτάχυνσης της βαρύτητας στο επιταχυνσιόμετρο και δυσκολίας στην απάλειψή της από τις μετρήσεις, χρησιμοποιείται μόνο η μέτρηση από το γυροσκόπιο. Στο ακόλουθο σχήμα παρουσιάζονται μετρήσεις και στους 3 άξονες από τα γυροσκόπια, τα επιταχυνσιόμετρα και μετά από τον συνδυασμό τους σε γωνίες από -90 έως 90 μοίρες.



Σχήμα 48 Μετρήσεις Επιταχυνσιόμετρου



Σχήμα 49 Μετρήσεις Γυροσκοπίου



Σχήμα 50 Φιλτραρισμένες Μετρήσεις

Η πλακέτα αδρανειακής μονάδας μέτρησης που χρησιμοποιήθηκε στην παρούσα πτυχιακή, ήταν η MPU-6050 με τα εξής χαρακτηριστικά ανά συσκευή:

Χαρακτηριστικά Γυροσκοπίου:

- Ψηφιακές έξοδοι γωνιακής ταχύτητας σε 3 άξονες (X,Y,Z) με προγραμματιζόμενη κλίμακα σε εύρος +250,+500,+1000,+2000 μοίρες ανά δευτερόλεπτο
- Εξωτερικός συγχρονισμός σήματος για την δυνατότητα συγχρονισμού με εικόνα, βίντεο και GPS.
- Ενσωματωμένος μετατροπέας αναλογικού σε ψηφιακό σήμα 16-bit με δυνατότητα ταυτόχρονης δειγματοληψίας από το γυροσκόπιο
- Ενισχυμένη σταθερότητα και σταθερή ευαισθησία σε θερμοκρασία καταστεί μειωμένη ανάγκη για βαθμονόμηση από τον χρήστη
- Καλύτερη λειτουργία σε μικρής συχνότητας θόρυβο
- Ψηφιακά προγραμματιζόμενο χαμηλοπερατό φίλτρο
- Κατανάλωση ρεύματος γυροσκοπίου: 3.6mA
- Ρεύμα ανάπαυσης: 5μΑ
- Εργοστασιακά βαθμονομημένη ευαισθησία συντελεστή κλίμακας
- Αυτοδιαγνωστικό τεστ οριζόμενο από τον χρήστη

Χαρακτηριστικά Επιταχυνσιόμετρου:

- Ψηφιακή έξοδος επιτάχυνσης σε 3 άξονες με προγραμματιζόμενη κλίμακα σε εύρος +2g, +4g, +8g, +16g
- Ενσωματωμένος μετατροπέας αναλογικού σε ψηφιακό σήμα 16-bit με δυνατότητα ταυτόχρονης δειγματοληψίας χωρίς την χρήση εξωτερικού πολυπλέκτη από το επιταχυνσιόμετρο
- Ονομαστικό ρεύμα επιταχυνσιόμετρου:500μΑ
- Μικρής ισχύος λειτουργία με ρεύμα:10μΑ στα 1.25Hz, 20μΑ στα 5Hz, 60μΑ στα 20Hz, 110μΑ στα 40Hz
- Ανίχνευση προσανατολισμού και σηματοδότηση
- Ανίχνευση πατήματος
- Interrupts προγραμματιζόμενα από τον χρήστη
- Υψηλό G interrupt
- Αυτοδιαγνωστικό τεστ οριζόμενο από τον χρήστη

Πρόσθετα χαρακτηριστικά του αισθητηρίου:

- Βοηθητικός master σε I2C δίαυλο για την ανάγνωση δεδομένων από εξωτερικούς αισθητήρες (π.χ. μαγνητόμετρο)
- 3.9mA ρεύμα λειτουργίας όταν και οι 6 άξονες μετριοούνται και ο επεξεργαστής δεδομένων είναι ενεργοποιημένος
- VDD τάση λειτουργίας εύρους 2.375V-3.46V
- Ο 1024 byte FIFO buffer μειώνει την κατανάλωση ισχύος με το να αφήνει τον επεξεργαστή να διαβάζει ριπές δεδομένων και να γυρνά σε λειτουργία μικρής κατανάλωσης

- Προγραμματιζόμενα από τον χρήστη ψηφιακά φίλτρα για το γυροσκόπιο, το επιταχυνσιόμετρο και τον αισθητήρα θερμοκρασίας
- Ανεκτικό σε σοκ έως 10,000g
- 400kHz γρήγορης λειτουργίας I2C για επικοινωνία με όλους τους καταχωρητές
- 8000Hz μέγιστη δειγματοληψία από το γυροσκόπιο
- 1000Hz μέγιστη δειγματοληψία από το επιταχυνσιόμετρο



Σχήμα 51 MPU-6050

3 Στρατηγικές Ελέγχου της ιπτάμενης πλατφόρμας

3.1 Εισαγωγή στον έλεγχο της ιπτάμενης πλατφόρμας

Το σύστημα της ιπτάμενης πλατφόρμας είναι ένα ασταθές σύστημα, το οποίο για να ελεγχθεί θα πρέπει να υλοποιηθεί ελεγκτής με ανάδραση για την εξισορρόπηση του συστήματος. Στο κεφάλαιο αυτό αναλύονται τα στάδια της στρατηγικής ελέγχου της πλατφόρμας, με σκοπό τον έλεγχο της γωνιακής ταχύτητας και θέσης της πλατφόρμας ανά άξονα. Με την χρήση εργαλείων εύρεσης της συνάρτησης μεταφοράς του συστήματος, μπορούν να πραγματοποιηθούν προσομοιώσεις για τον υπολογισμό των μεταβλητών των ελεγκτών[A3]. Για την υλοποίηση των ελεγκτών, η πλατφόρμα τοποθετήθηκε στη διάταξη που παρουσιάζεται στο ακόλουθο σχήμα, με σκοπό την απομονωμένη λειτουργία ανά άξονα στους άξονες X και Y. Για την υλοποίηση του ελεγκτή του Z άξονα κατασκευάστηκε διάταξη που στερεώνεται η πλατφόρμα σε άξονα και με χρήση ρουλεμάν στρέφεται χωρίς μεγάλη τριβή.



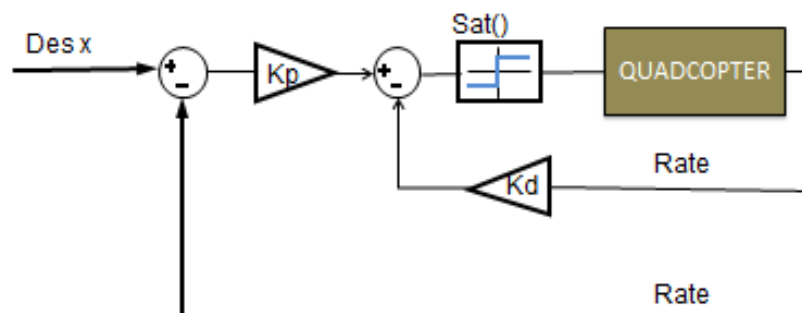
3.2 Στρατηγικές ελέγχου της ιπτάμενης πλατφόρμας.

Έλεγχος στους X, Y άξονες

Οι ελεγκτές για τους X, Y (pitch και roll αντίστοιχα) άξονες είναι ίδιοι λόγω ισόκατανομής της πλατφόρμας. Σημειώνεται ότι, για τον έλεγχο ανά άξονα χρησιμοποιούνται οι δύο απέναντι ομόρροποι μεταξύ τους κινητήρες. Ο έλεγχος της γωνιακής θέσης απαιτεί τον έλεγχο της γωνιακής ταχύτητας, χωρίς τον οποίο δεν είναι δυνατός. Για το λόγο αυτό

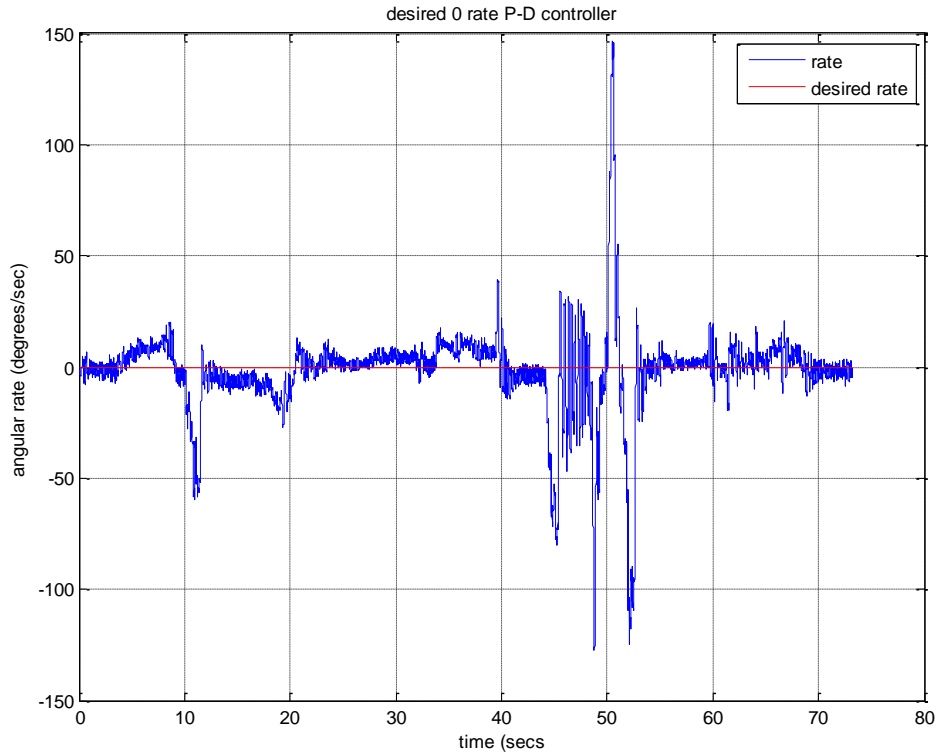
χρησιμοποιήθηκαν δύο βρόγχοι με ανάδραση, ένας για τον έλεγχο της γωνιακής ταχύτητας και εξωτερικά του για τον έλεγχο γωνιακής θέσης. Από το αισθητήριο μετρούνται, η γωνιακή ταχύτητα από το γυροσκόπιο και η γωνία θέσης από το complimentary φίλτρο. Στα πειράματα που πραγματοποιήθηκαν, διαφοροποιούνται οι επιθυμητές τιμές των γωνιακών ταχυτήτων και γωνιακών θέσεων για να διακρίνεται περισσότερο η απόκριση του συστήματος. Επίσης δοκιμάστηκαν και παρεμβολές για τον έλεγχο της σωστής λειτουργίας του εκάστοτε ελεγκτή. Αρχικά για τον έλεγχο γωνιακής ταχύτητας υλοποιήθηκε ελεγκτής P-D, με επιθυμητή μηδενική γωνιακή ταχύτητα, του οποίου το διάγραμμα παρουσιάζεται παρακάτω.

P-D controller diagram



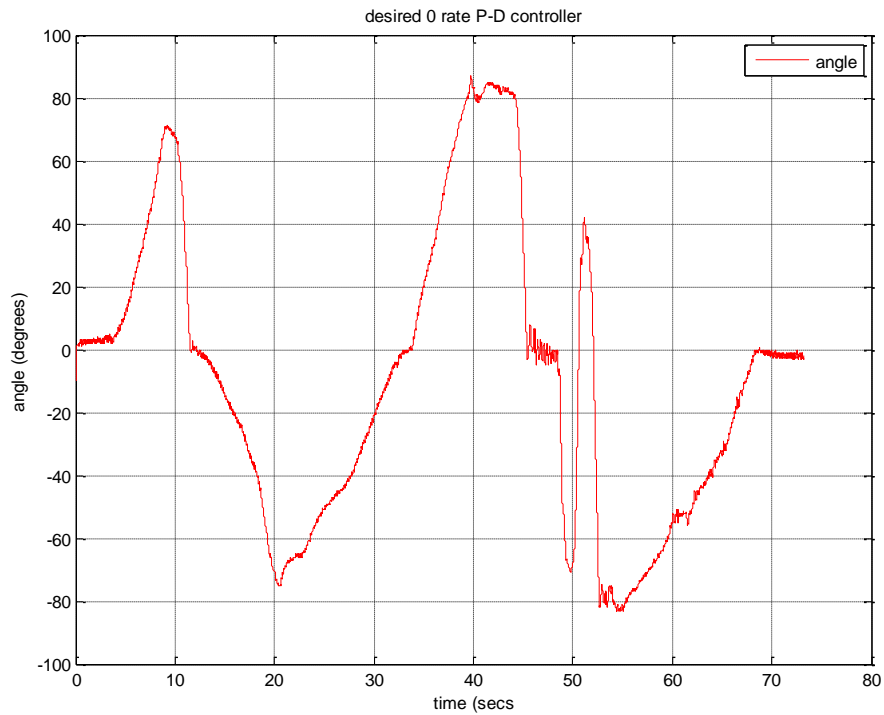
Σχήμα 52 Διάγραμμα ελεγκτή Γωνιακής ταχύτητας X άξονα

Τα κέρδη του ελεγκτή προέκυψαν από πειραματικές δοκιμές και οι τιμές τους είναι **$K_p=1.2$** **$K_d=3$** . Η απόκριση του ελεγκτή εικονίζεται στο ακόλουθο σχήμα:

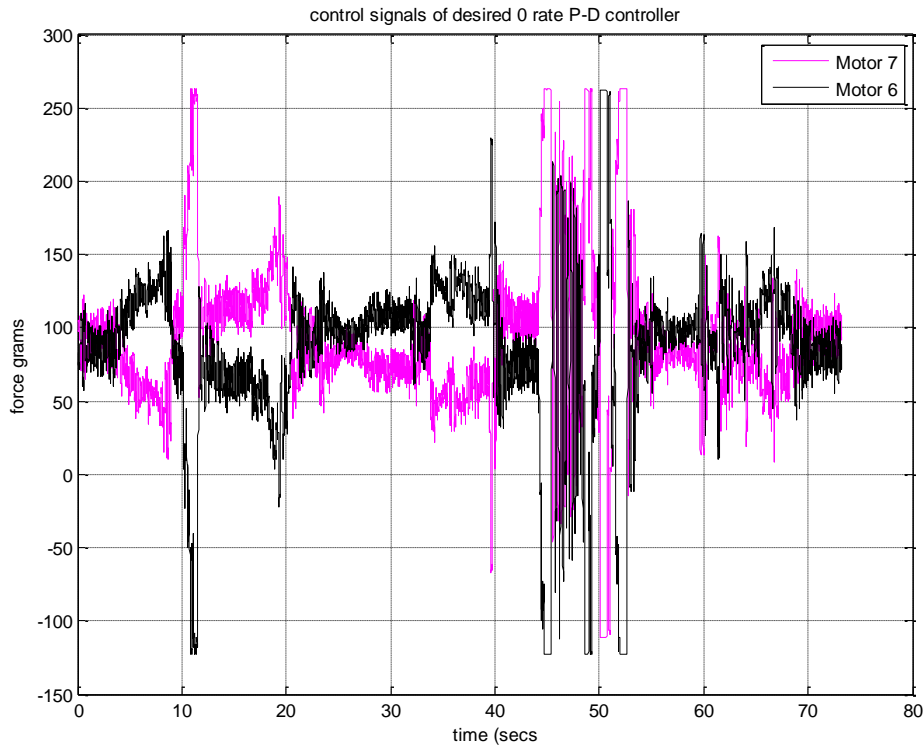


Σχήμα 53 Η απόκριση του συστήματος με παρεμβολές προς τις δύο κατευθύνσεις.

Παρατηρείται ότι λόγω βαρυτικών δυνάμεων και δονήσεων από τους κινητήρες η απόκριση δεν μηδενίζεται σταθερά. Η απόκριση της γωνιακής θέσης καθώς και τα σήματα ελέγχου για κάθε κινητήρα κατά την διάρκεια του πειράματος ικονίζονται παρακάτω.



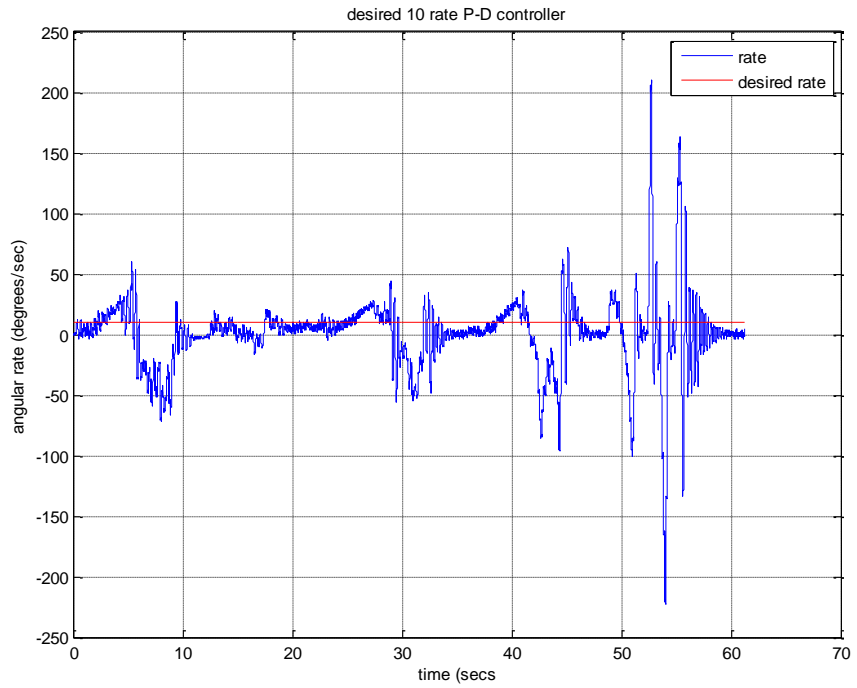
Σχήμα 54 Απόκριση γωνιακής θέσης του ελεγκτή γωνιακής ταχύτητας



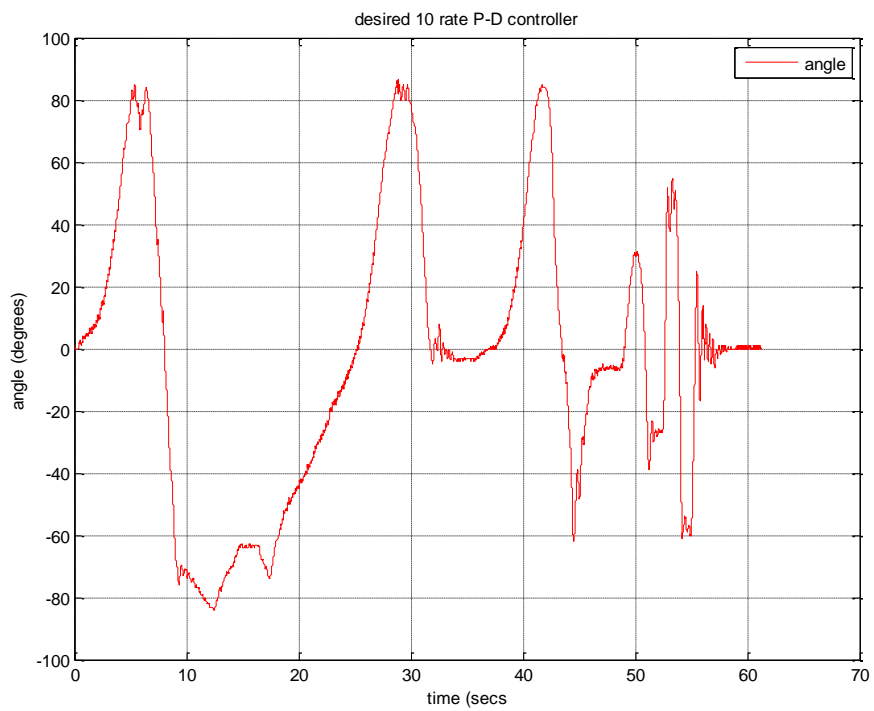
Σχήμα 55 Σήματα ελέγχου των κινητήρων για τον ελεγκτή γωνιακής ταχύτητας

Στο σχήμα παρατηρείται η αναλογική σχέση μεταξύ των δύο κινητήρων, όπου όταν αυξάνεται το σήμα ελέγχου για τον ένα μειώνεται ακριβώς ίσα και αντίθετα για τον άλλο. Επίσης παρατηρείται στις ελάχιστες και μέγιστες τιμές το όριο που υπάρχει με σκοπό την αποτροπή κορεσμού των σημάτων ελέγχου.

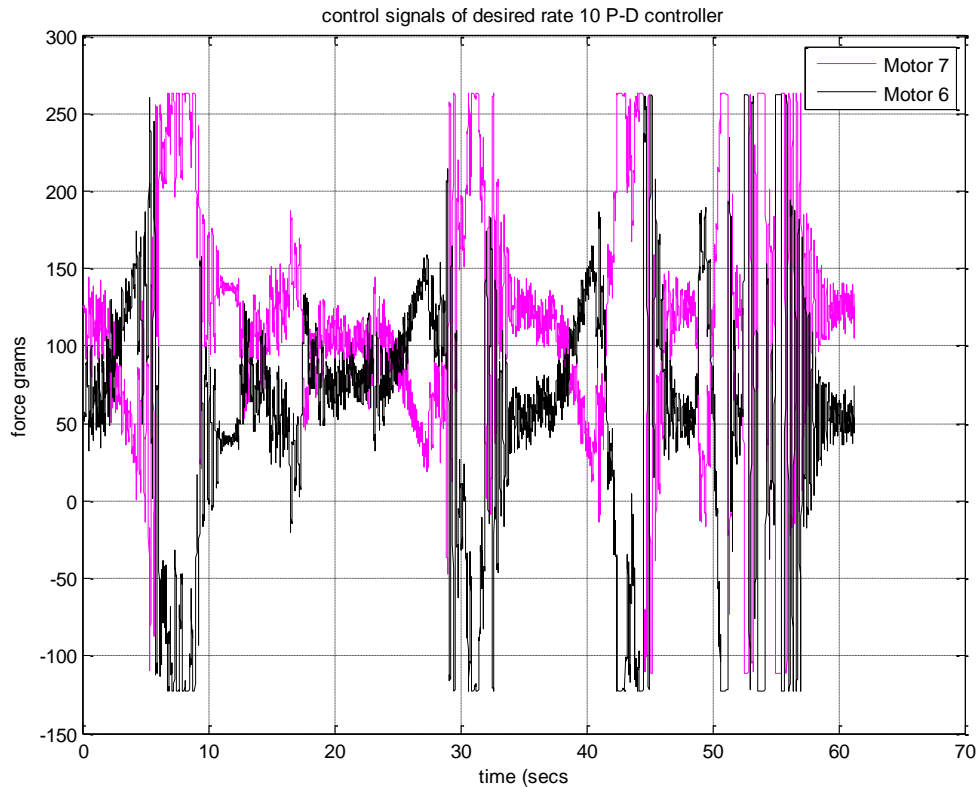
Τέλος, για τον ελεγκτή γωνιακής ταχύτητας πραγματοποιήθηκε πείραμα με επιθυμητή γωνιακή ταχύτητα $10^\circ/\text{sec}$. Η απόκριση της γωνιακής ταχύτητας, η απόκριση της γωνιακή θέσης της πλατφόρμας και τα σήματα ελέγχου παρουσιάζονται στα ακόλουθα σχήματα.



Σχήμα 56 Απόκριση ελεγκτή γωνιακής θέσης με επιθυμητή γωνιακή ταχύτητα 10deg/sec



Σχήμα 57 Απόκριση γωνιακής θέσης του ελεγκτή γωνιακής ταχύτητας

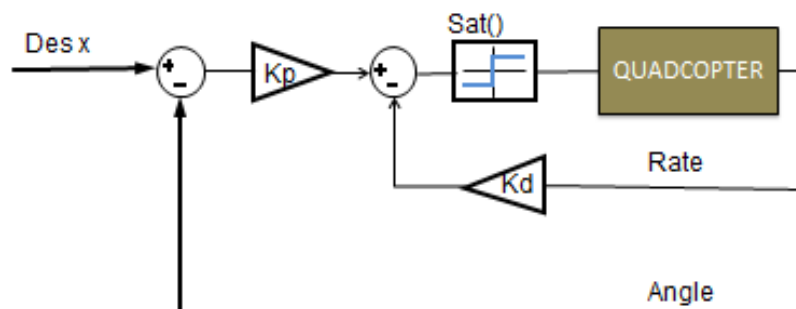


Σχήμα 58 Σήματα ελέγχου των κινητήρων για τον ελεγκτή γωνιακής ταχύτητας

Από τις αποκρίσεις παρατηρείται, απότομη συμπεριφορά σε σχέση με την επιθυμητή γωνιακή θέση 0deg/sec του προηγούμενου πειράματος, ιδιαίτερα στα σημεία που υπήρχε παρεμβολή στο σύστημα.

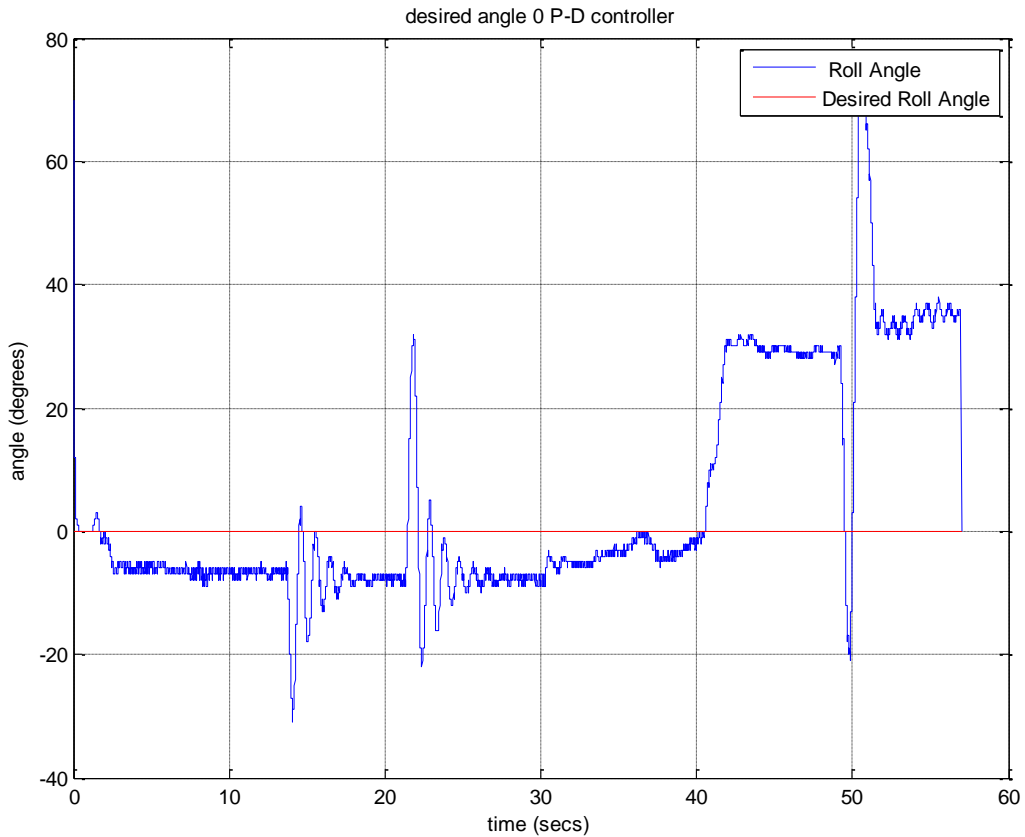
Στην συνέχεια με την επίτευξη του ελέγχου γωνιακής ταχύτητας υλοποιήθηκε ελεγκτής P-D για τον έλεγχο γωνιακής θέσης του οποίου το διάγραμμα είναι το εξής:

P-D controller diagram



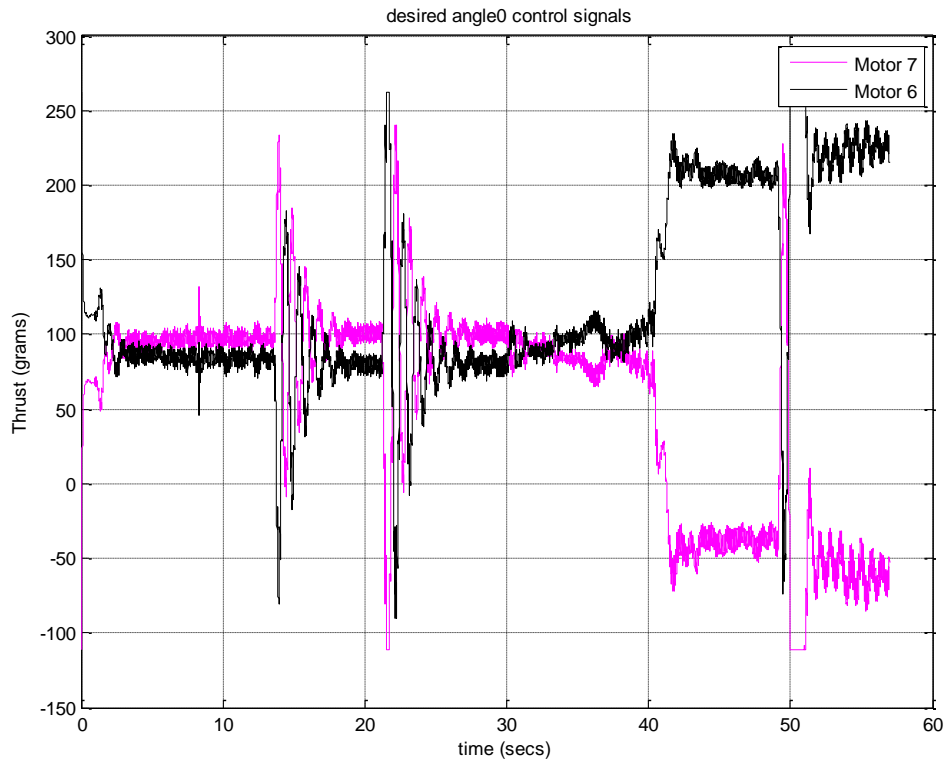
Σχήμα 59 Διάγραμμα γωνιακής θέσης P-D ελεγκτή X άξονα

Τα κέρδη των ελεγκτών βρέθηκαν πειραματικά και οι τιμές τους είναι $K_p=2.5$ $K_d=2.8$. Στο ακόλουθο σχήμα εικονίζεται η απόκριση του συστήματος με P-D ελεγκτή.



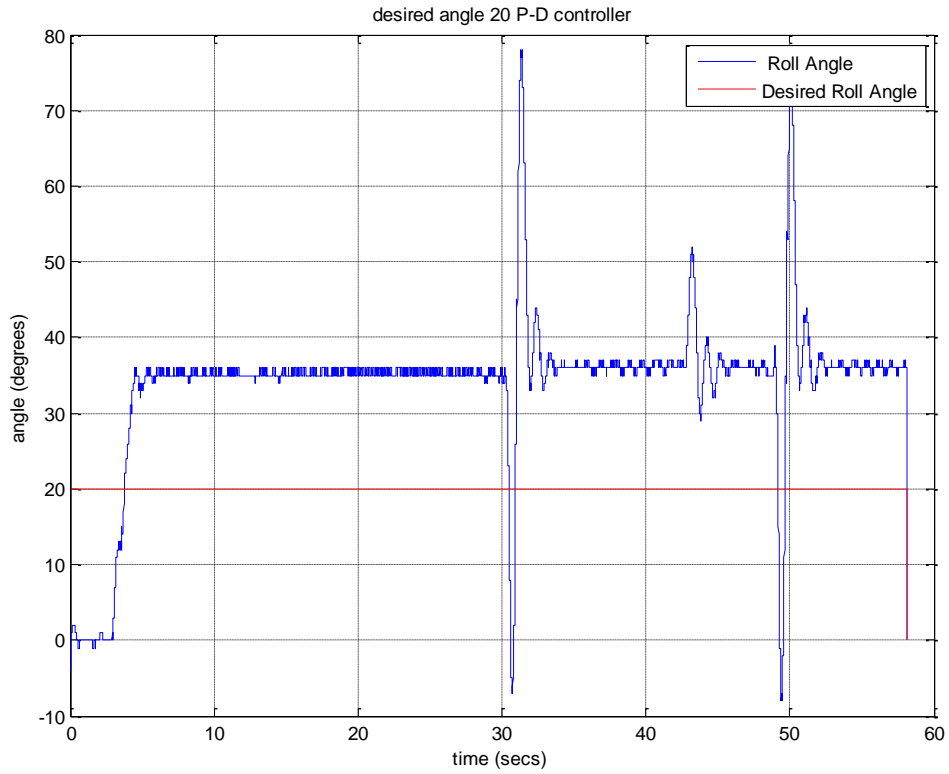
Σχήμα 60 Απόκριση γωνιακής θέσης P-D ελεγκτή

Στο παραπάνω σχήμα παρατηρείται, για επιθυμητή γωνία τις 0 μοίρες σφάλμα μόνιμης κατάστασης. Το σφάλμα οφείλεται στην ανομοιότητα της ώσης των κινητήρων και εξαλείφεται προσωρινά, με χρήση αντιστάθμισης των τιμών στα σήματα ελέγχου των κινητήρων. Επίσης υπάρχουν παρεμβολές για την παρατήρηση της επαναφοράς της πλατφόρμας στην επιθυμητή θέση. Τέλος, τοποθετήθηκε μόνιμη παρεμβολή η οποία ξεκινάει περίπου από τα 40 δευτερόλεπτα, και πραγματοποιήθηκε μία στιγμιαία παρεμβολή ενώ υπήρχε η μόνιμη παρεμβολή στο σύστημα με σκοπό, την διερεύνηση δυνατότητας επαναφοράς της πλατφόρμας στην προηγούμενη θέση. Στο ακόλουθο σχήμα παρουσιάζονται τα σήματα ελέγχου των κινητήρων για τον ελεγκτή γωνιακής θέσης:

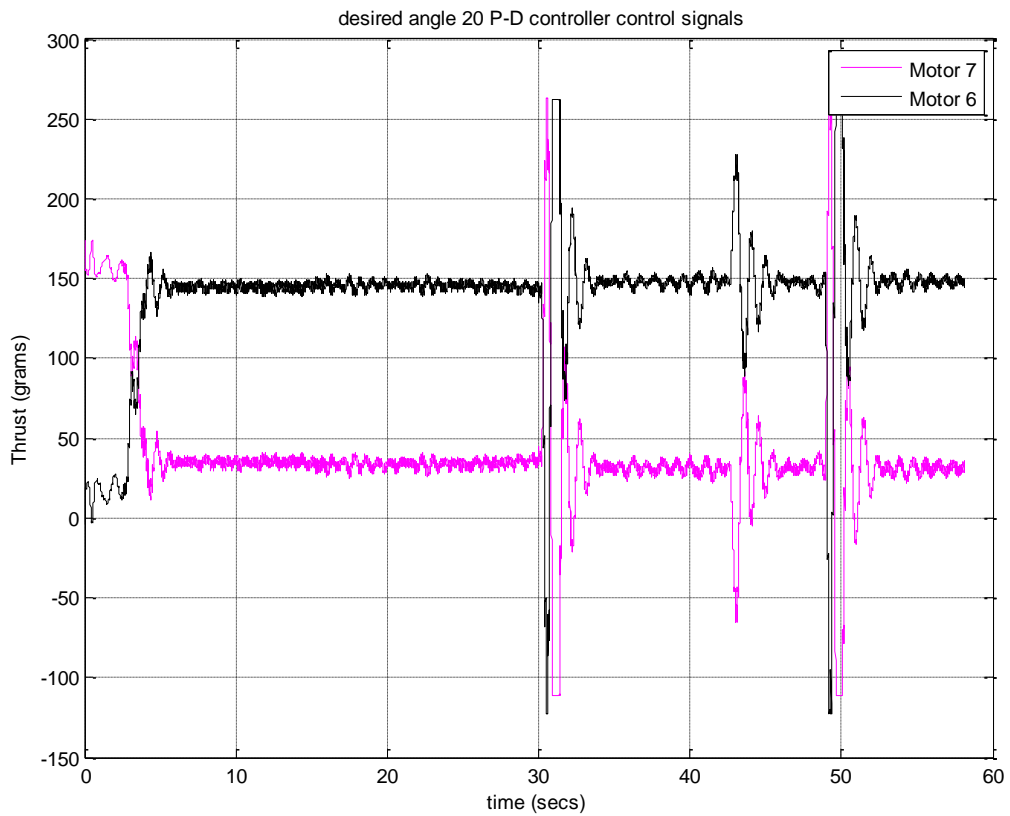


Σχήμα 61 Σήματα Ελέγχου του P-D ελεγκτή γωνιακής θέσης

Στην συνέχεια πραγματοποιήθηκε πείραμα με επιθυμητή γωνιακή θέση τις 20 μοίρες και στα ακόλουθα σχήματα φαίνονται, η απόκριση της γωνιακής θέσης του συστήματος και τα σήματα ελέγχου των κινητήρων.



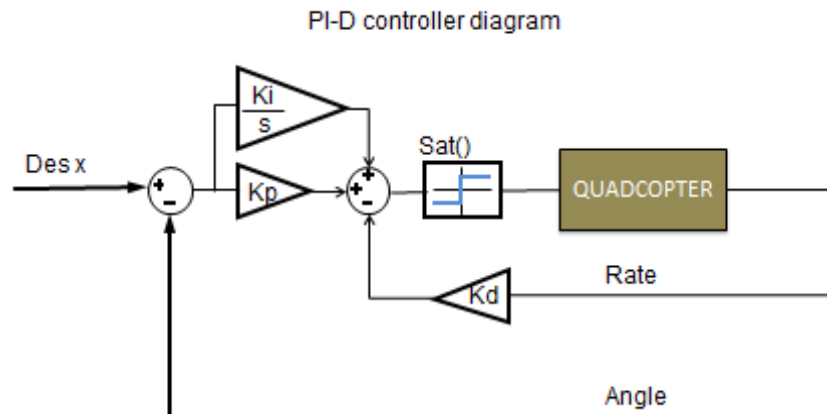
Σχήμα 62 Απόκριση γωνιακής θέσης P-D ελεγκτή για επιθυμητή γωνιακή θέση 20 μοίρες



Σχήμα 63 Σήματα ελέγχου των κινητήρων

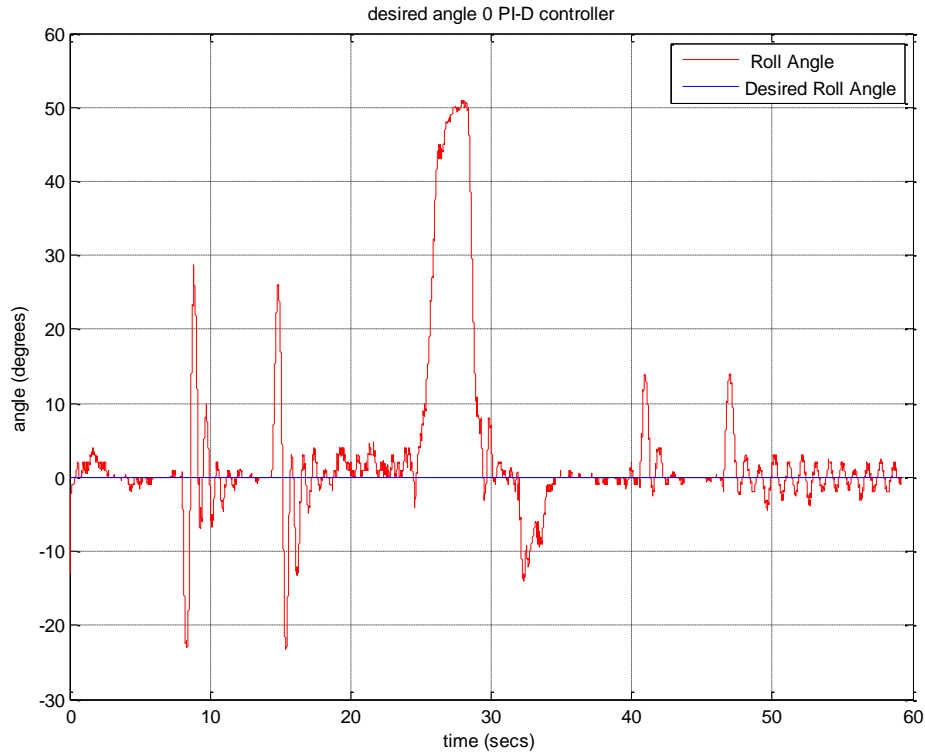
Από την απόκριση της γωνιακής θέσης παρατηρείται, αυξημένο σφάλμα μόνιμης κατάστασης λόγω της μεγάλης γωνιακής θέσης της πλατφόρμας. Η αύξηση της επιθυμητής γωνιακής θέσης έχει ως αποτέλεσμα, την αύξηση της επιρροής των βαρυτικών δυνάμεων που παρεμβαίνουν στο σύστημα, όπου σε συνδυασμό με το όριο των κινητήρων το σφάλμα μόνιμης κατάστασης αυξάνεται.

Με τον ελεγκτή P-D επιτεύχθηκε ο έλεγχος γωνιακής θέσης με μόνο μειονέκτημα το σφάλμα μόνιμης κατάστασης, το οποίο για να εξαλειφθεί προστέθηκε I όρος, με όριο του σήματος ελέγχου το 10% του ολικού εύρους λειτουργίας των κινητήρων για την αποφυγή κορεσμού. Το διάγραμμα του τελικού PI-D ελεγκτή είναι το ακόλουθο:



Σχήμα 64 Διάγραμμα γωνιακής θέσης PI-D ελεγκτή X άξονα

Τα κέρδη των PI-D είναι τα εξής **Kp=2.8 Ki=2.5 Kd=1** και είναι κοινά στους δύο άξονες. Η απόκριση του ελεγκτή στον Y άξονα (Roll) για επιθυμητή γωνιακή θέση 0 μοίρες παρουσιάζεται παρακάτω στο σχήμα, με μόνιμη και στιγμιαίες παρεμβολές.



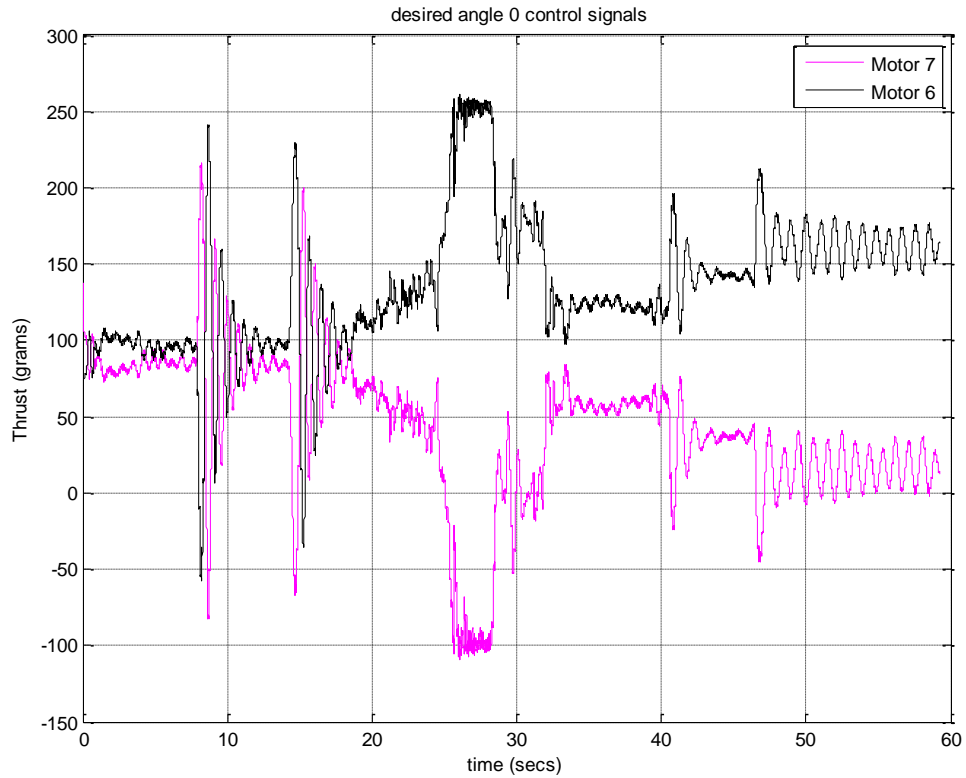
Σχήμα 65 Απόκριση γωνιακής θέσης PI-D ελεγκτή για επιθυμητή γωνιακή θέση τις 0 μοίρες

Παρατηρείται ότι με την προσθήκη του I όρου, εξαλείφεται το σφάλμα μόνιμης κατάστασης ακόμα και με την προσθήκη μόνιμης παρεμβολής. Σημειώνεται ότι, τα όρια των σημάτων ελέγχου των κινητήρων για την χρήση τους σε διορθωτικές κινήσεις είναι στο 10% της ολικής ώσης των κινητήρων. Αυξομειώνοντας αυτά τα όρια υπάρχουν δύο περιπτώσεις:

Αυξάνοντας τα όρια, αυξάνεται η σταθερότητα σε παρεμβολές (μεγαλύτερους ανέμους ή προσθήκη μόνιμων παρεμβολών (gimbal,βραχίονες,κλι)) και μειώνεται το ποσοστό που χρησιμοποιείται για αλλαγή υψομέτρου

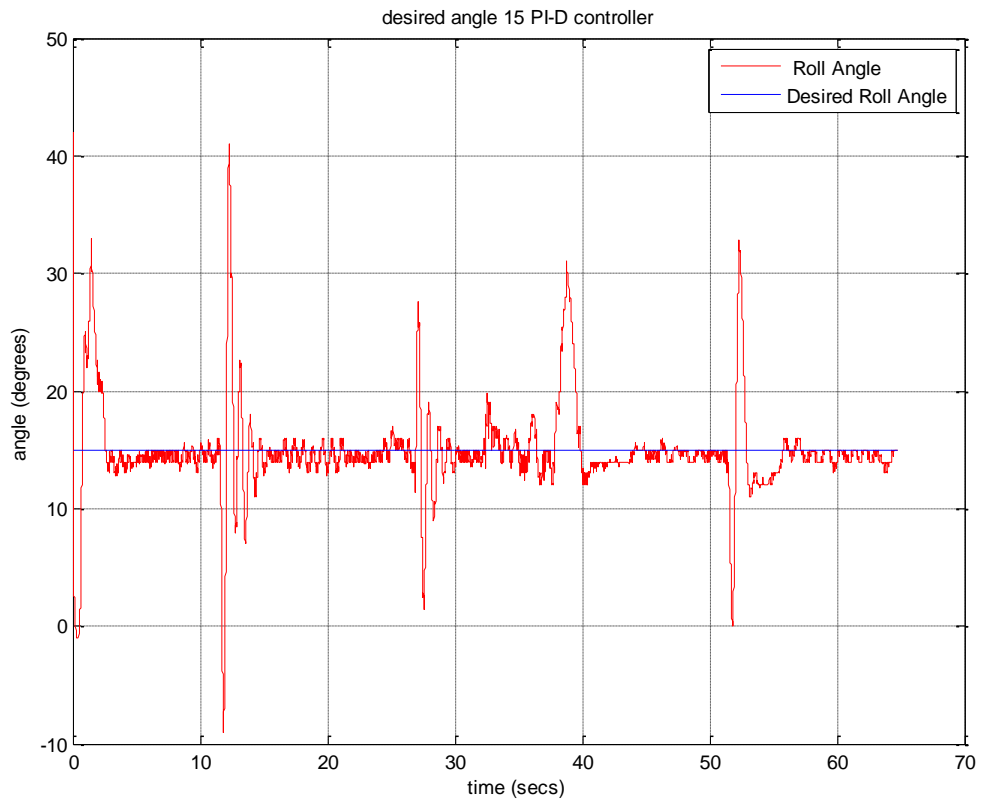
Μειώνοντας τα όρια, μειώνεται η σταθερότητα σε παρεμβολές, με κίνδυνο ακόμα και την πτώση σε ακραίες περιπτώσεις, και αυξάνεται το ποσοστό που χρησιμοποιείται για αλλαγή υψομέτρου.

Στο ακόλουθο σχήμα εικονίζονται τα σήματα ελέγχου των κινητήρων του ελεγκτή.

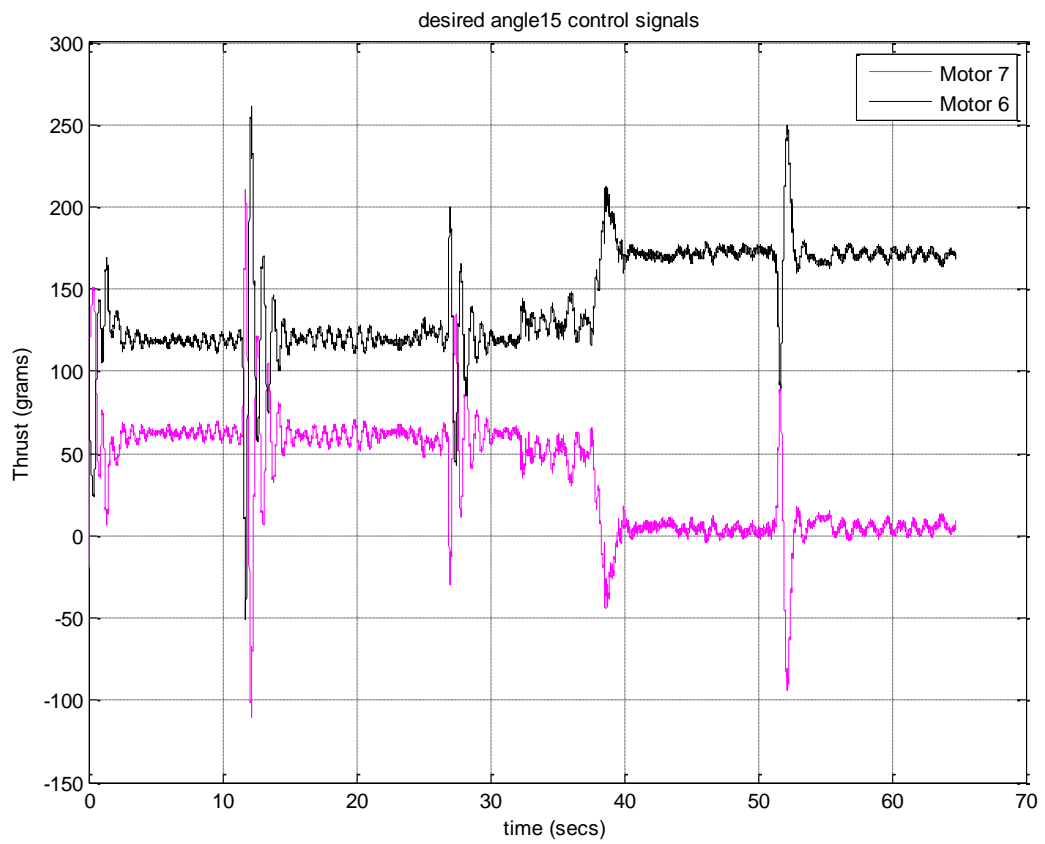


Σχήμα 66 Σήματα ελεγκτή για τους κινητήρες

Στο επόμενο πείραμα, με επιθυμητή γωνιακή θέση τις 15 μοίρες, στιγμιαίες και μόνιμες παρεμβολές, η απόκριση και τα σήματα ελέγχου παρουσιάζονται στα ακόλουθα σχήματα. Παρατηρείται ότι, με την αύξηση της επιθυμητής γωνιακής θέσης η συμπεριφορά του ελεγκτή δεν μεταβλήθηκε, και το σύστημα παραμένει ευσταθές.

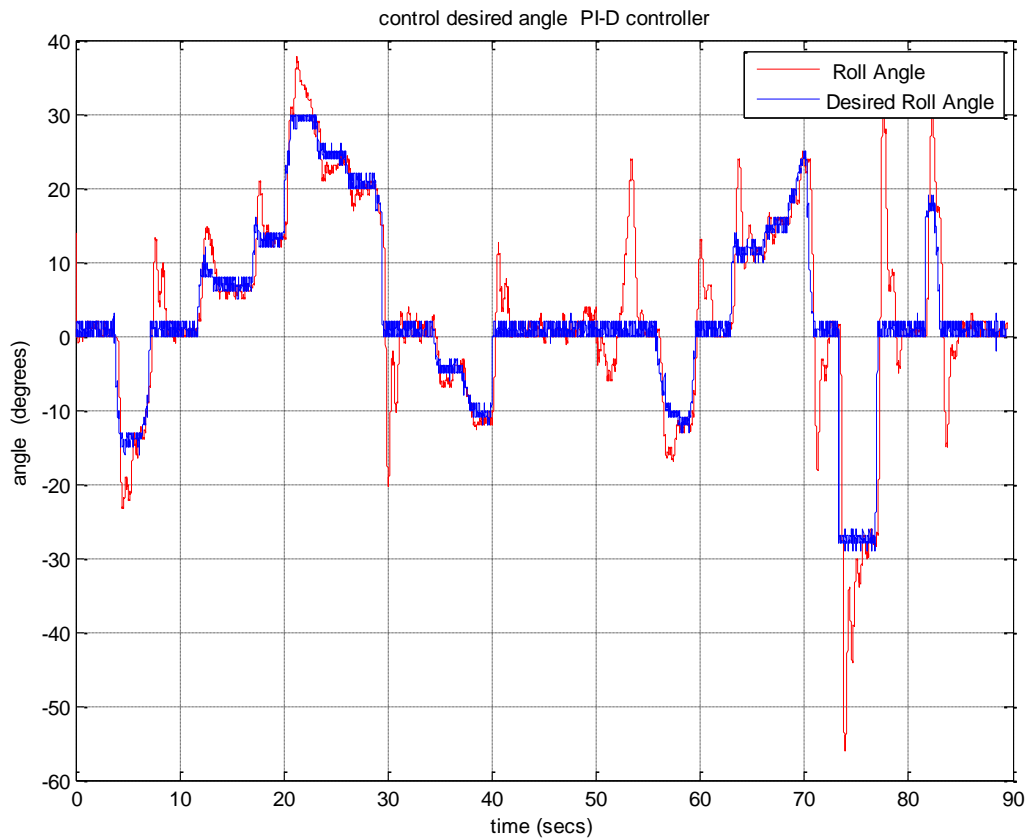


Σχήμα 67 Απόκριση γωνιακής θέσης PI-D ελεγκτή με επιθυμητή γωνιακή θέση τις 15 μοίρες

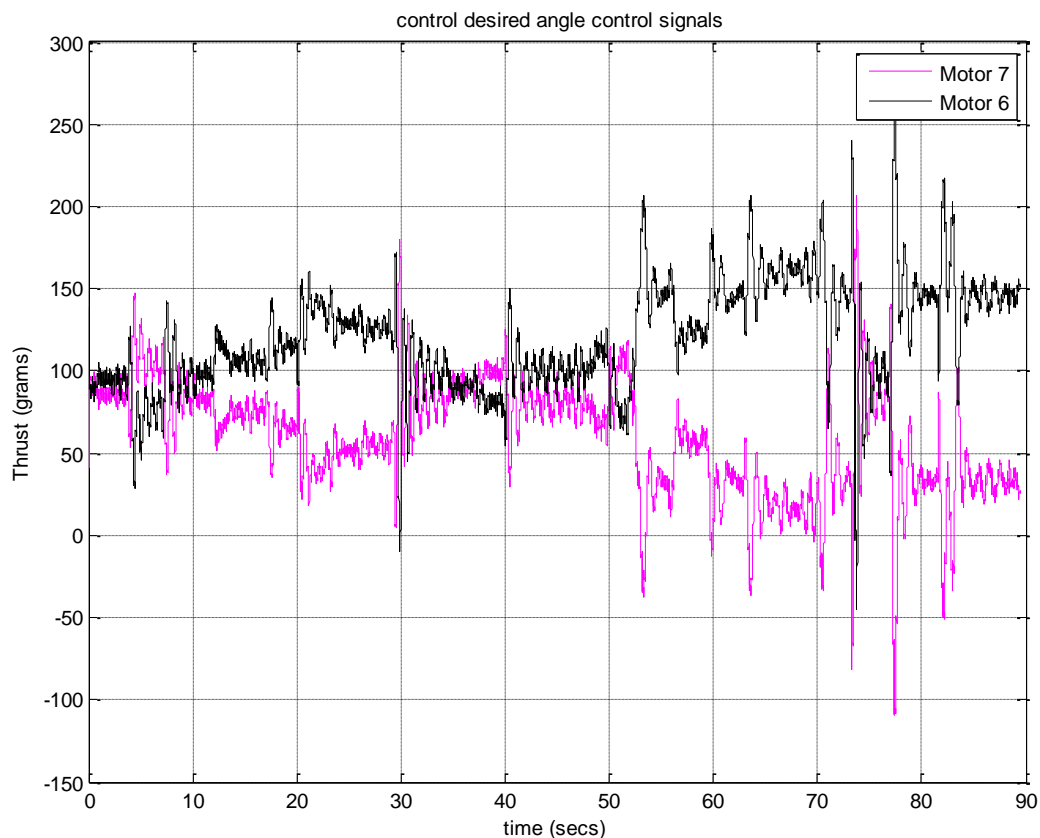


Σχήμα 68 Σήματα του ελεγκτή για τους κινητήρες

Τέλος, για να παρατηρηθεί καλύτερα η συμπεριφορά του συστήματος με τον PI-D ελεγκτή, χρησιμοποιήθηκε η τηλεκατεύθυνση για τον ορισμό της επιθυμητής γωνιακής θέσης, η οποία μεταβαλλόταν με όρια τις 30, -30 μοίρες. Στα ακόλουθα σχήματα εικονίζονται η απόκριση του συστήματος και τα σήματα ελέγχου των κινητήρων.



Σχήμα 69 Απόκριση γωνιακής θέσης του PI-D ελεγκτή με την χρήση τηλεκατεύθυνσης



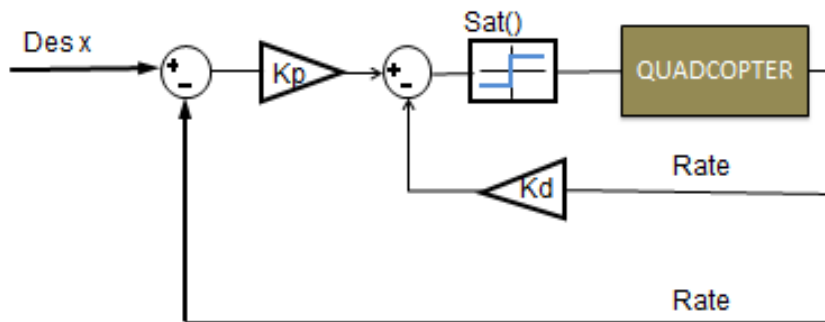
Σχήμα 70 Σήματα ελέγχου των κινητήρων από τον ελεγκτή

Από το σχήμα της απόκρισης της γωνιακής θέσης παρατηρείται υπερρύψωση, η οποία οφείλεται στην βαρύτητα λόγω αύξησης της γωνιακής θέσης. Σημειώνεται πως στο σχήμα της απόκρισης υπάρχουν στιγμιαίες και μόνιμες παρεμβολές. Τα κέρδη των ελεγκτών είναι κοινά στους άξονες X,Y (pitch και roll αντίστοιχα). Με χρήση των δύο ελεγκτών μόνο, δύνανται να πραγματοποιηθεί πτήση με μειωμένο έλεγχο.

Έλεγχος στον Z άξονα

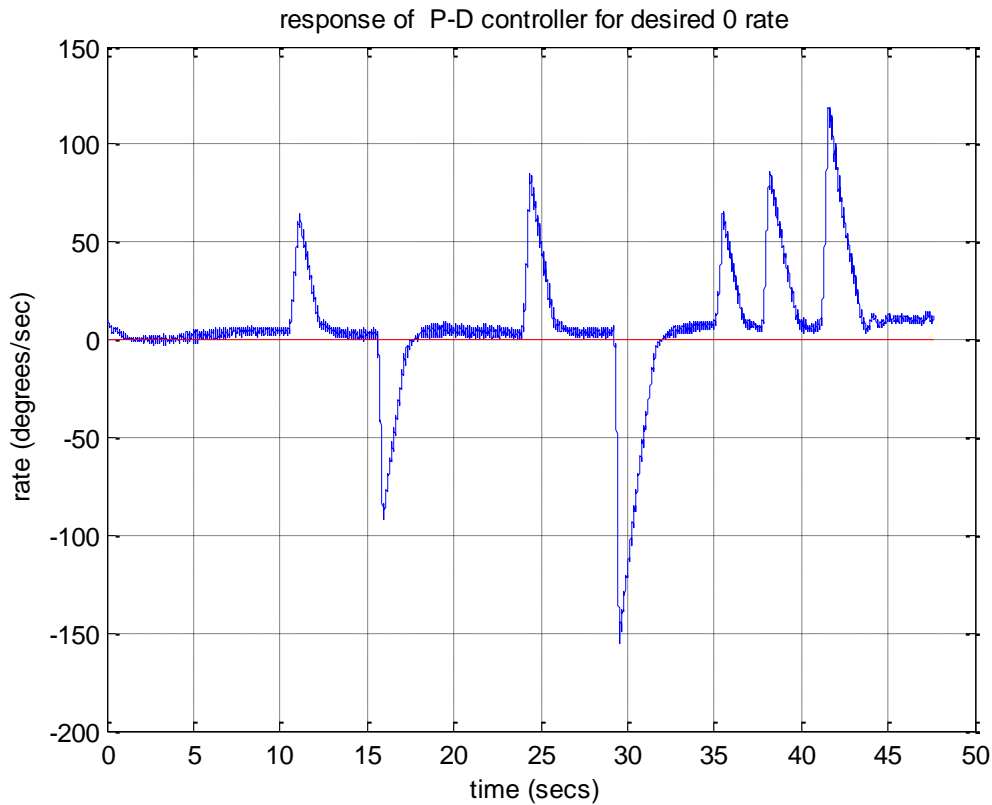
Για τον έλεγχο του Z άξονα είναι απαραίτητη η ρύθμιση και των τεσσάρων κινητήρων. Ο έλεγχος γωνιακής θέσης στον Z άξονα δεν είναι εφικτός, γιατί το αισθητήριο θέσης μετράει γωνία με μετάπτωση στον συγκεκριμένο άξονα, γεγονός που καθιστά αδύνατο τον έλεγχο γωνιακής θέσης. Για το λόγο, αυτό υλοποιήθηκαν μόνο οι ελεγκτές γωνιακής ταχύτητας. Πρώτα υλοποιήθηκε ελεγκτής P-D με το εξής διάγραμμα ελέγχου:

P-D controller diagram



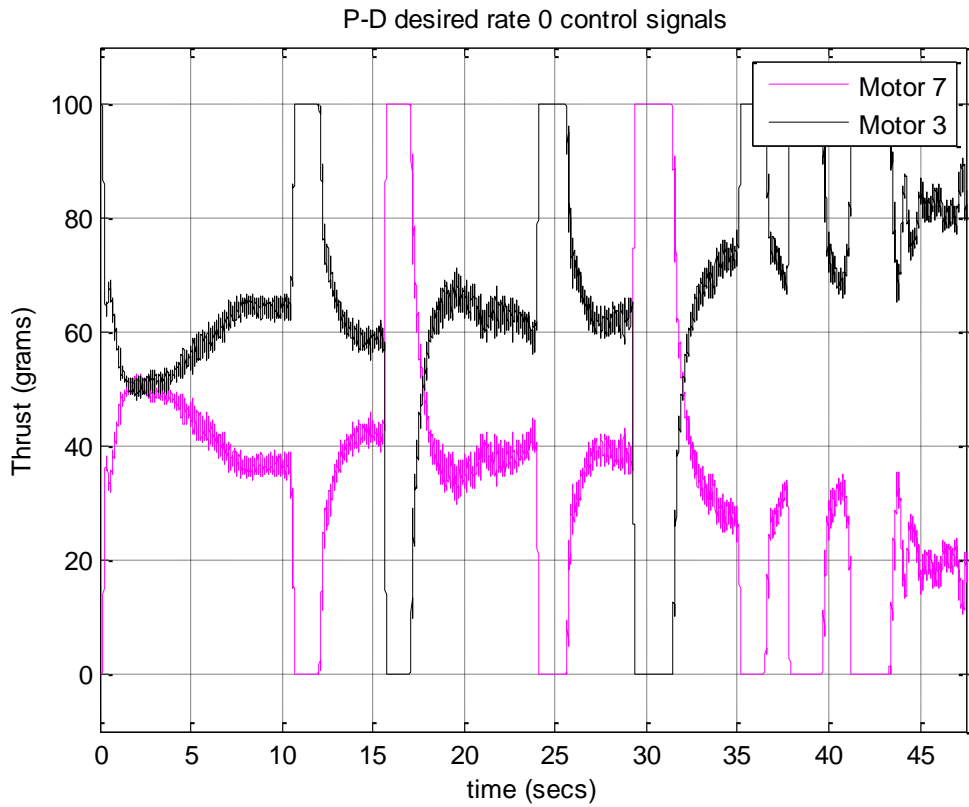
Σχήμα 71 Διάγραμμα ελέγχου γωνιακής ταχύτητας με P-D ελεγκτή

Η απόκριση και τα σήματα ελέγχου ανά κινητήρα του ελεγκτή παρουσιάζονται στα ακόλουθα σχήματα:

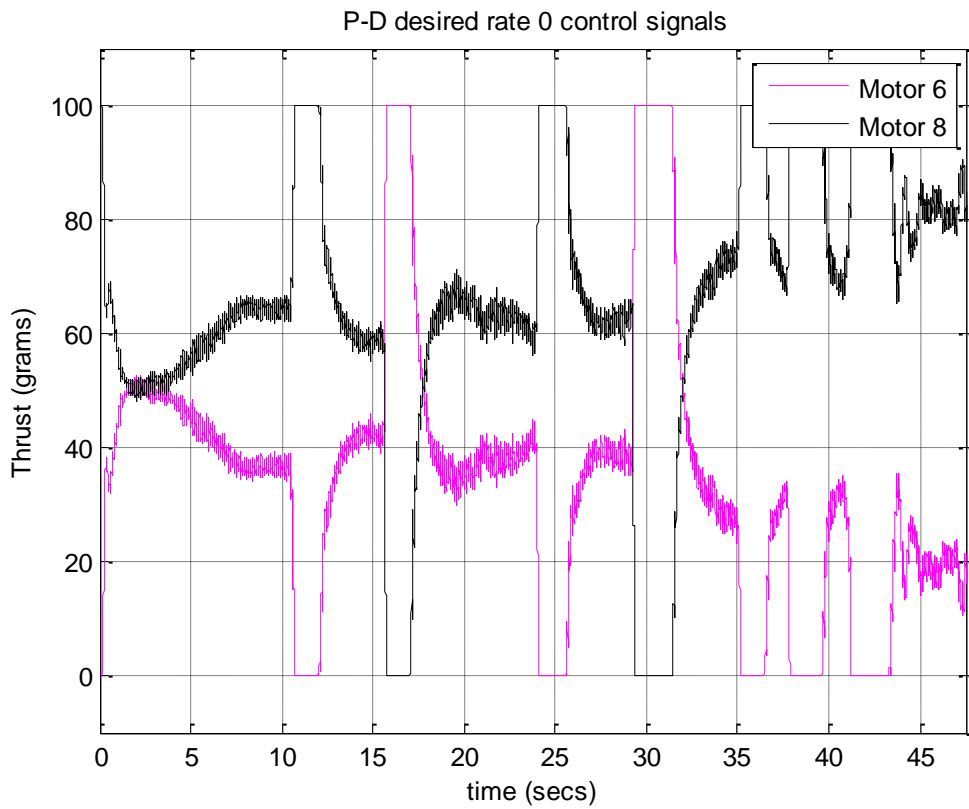


Σχήμα 72 Απόκριση γωνιακής ταχύτητας του P-D ελεγκτή για επιθυμητή γωνιακή ταχύτητα 0

Στο παραπάνω σχήμα παρατηρείται, η απόκριση του ελεγκτή με παρεμβολές σε τυχαίες χρονικές στιγμές και η επαναφορά της γωνιακής ταχύτητας σε μικρές τιμές. Σημειώνεται ότι, κατά την διάρκεια του πειράματος η γωνιακή θέση δεν καταγράφηκε.



Σχήμα 73 Σήματα ελέγχου κινητήρων 7 και 3

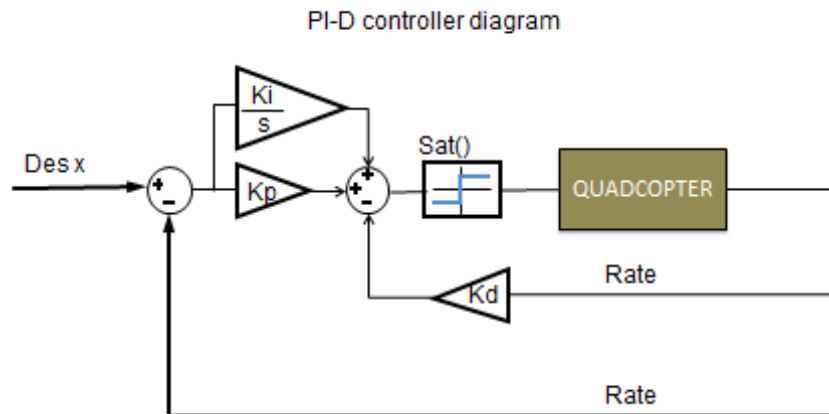


Σχήμα 74 Σήματα ελέγχου κινητήρων 6 και 8

Στα παραπάνω δύο σχήματα παρουσιάζονται τα σήματα ελέγχου για κάθε κινητήρα, με τους κινητήρες που στρέφονται αντίθετα μεταξύ τους να είναι στο ίδιο σχήμα. Επίσης παρατηρείται ότι, στις παρεμβολές που υπέστη το σύστημα τα δύο σήματα ελέγχου των κινητήρων αυξομειώνονται αντιστρόφως ανάλογα μεταξύ τους. Τέλος στις άκρες τους παρατηρείται η παρουσία ελέγχου του κορεσμού του σήματος, η οποία έχει τιμή ± 50 γραμμάρια δύναμης.

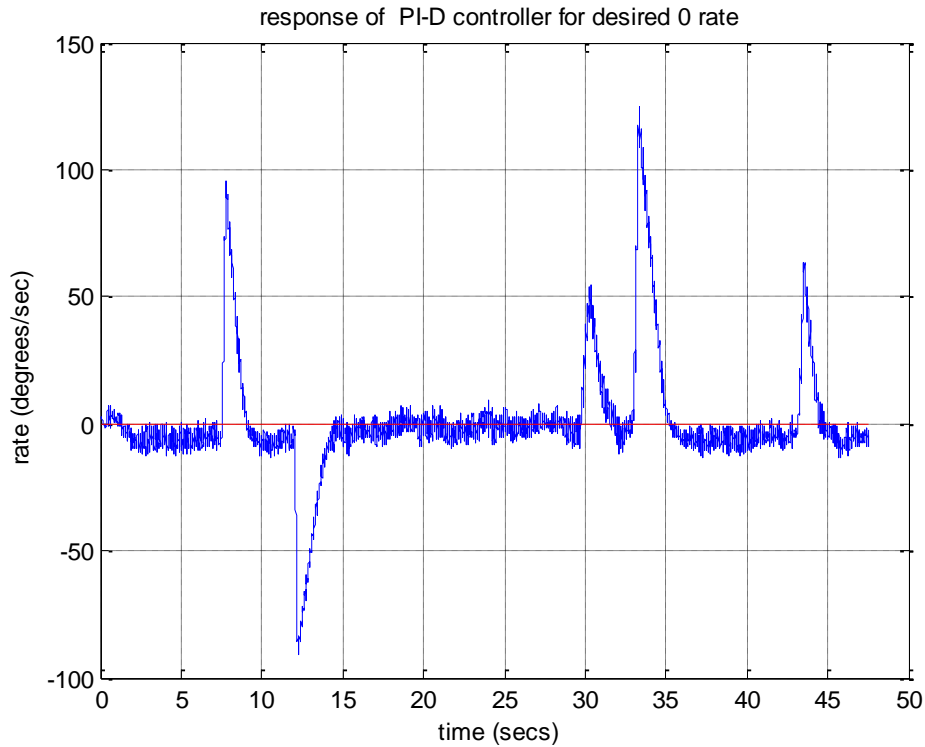
Τα κέρδη του ελεγκτή έχουν τιμές $K_p=10$, $K_d=2$.

Στη συνέχεια, υλοποιήθηκε ελεγκτής PI-D για τον έλεγχο γωνιακής ταχύτητας με το ακόλουθο διάγραμμα ελέγχου.



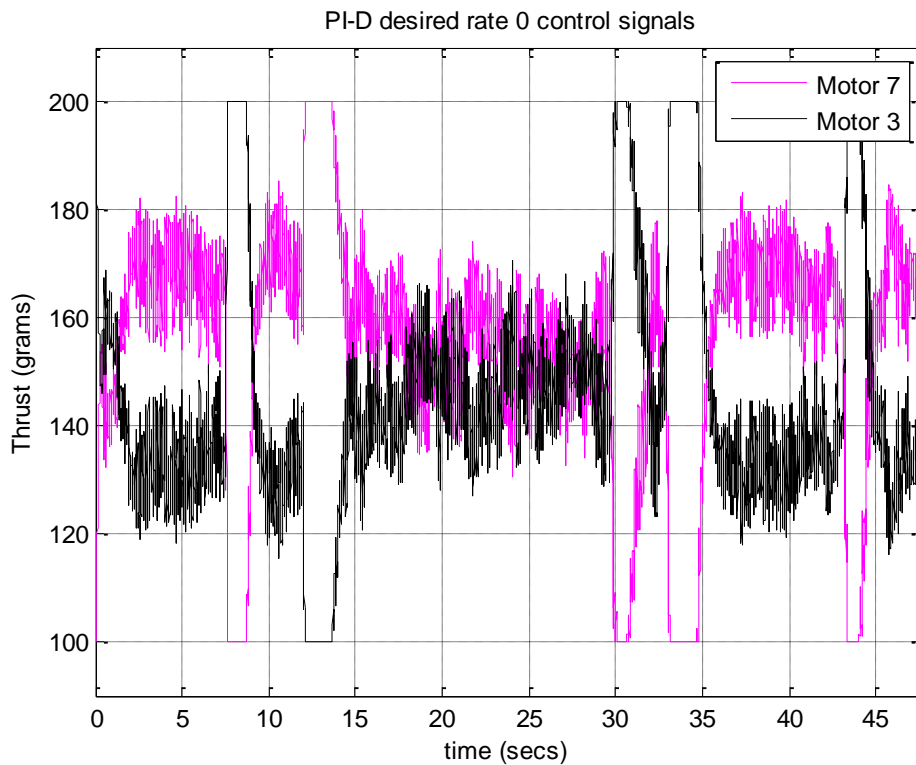
Σχήμα 75 Διάγραμμα ελέγχου PI-D ελεγκτή

Στα ακόλουθα σχήματα παρουσιάζονται η απόκριση του ελεγκτή και τα σήματα ελέγχου για κάθε κινητήρα.

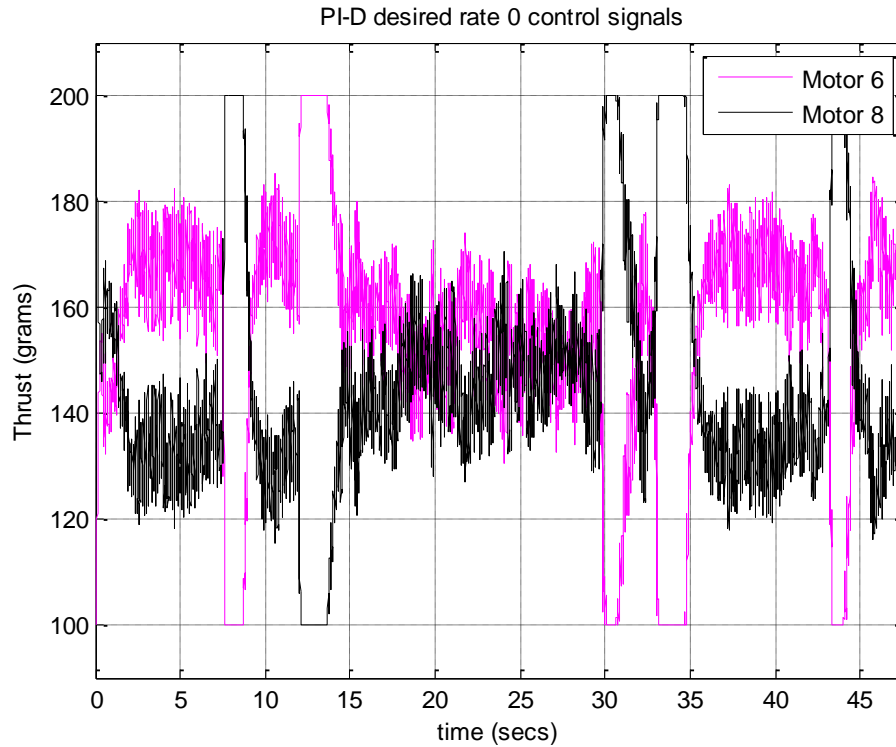


Σχήμα 76 Απόκριση PI-D ελεγκτή

Παρατηρείται ότι με τη προσθήκη του I όρου η απόκριση του ελεγκτή με παρεμβολές σε τυχαίες χρονικές στιγμές, έχει σαν αποτέλεσμα την ταχύτερη απόκριση και μείωση του σφάλματος μόνιμης κατάστασης σε ικανοποιητικό βαθμό.



Σχήμα 77 Σήματα ελέγχου των κινητήρων M7 και M3



Σχήμα 78 Σήματα ελέγχου των κινητήρων M6 και M8

Στα παραπάνω σχήματα παρουσιάζονται τα σήματα ελέγχου για κάθε ζευγάρι κινητήρων. Τα κέρδη του ελεγκτή έχουν τιμές $K_p=10$, $K_i=2.5$, $K_d=2$. Στον παρακάτω πίνακα παρουσιάζονται τα κέρδη όλων των ελεγκτών που υλοποιήθηκαν.

Αξονας	Έλεγκτής	P	I	D
X	Ταχύτητας	1.2	-	3
Y	Ταχύτητας	1.2	-	3
X	Θέσης	2.8	2.5	1
Y	Θέσης	2.8	2.5	1
Z	Ταχύτητας	10	-	2
Z	Ταχύτητας	10	2.5	2

Πίνακας 5 Κέρδη ελεγκτών

3.3 Ροή του προγράμματος ελέγχου

Η ροή του προγράμματος ελέγχου έχει δύο σκέλη, τα οποία είναι η αρχικοποίηση του προγράμματος και ο βρόγχος ελέγχου.

Στην αρχικοποίηση υπάρχουν τα στοιχεία εκείνα που τρέχουν για μία φορά κατά την εκκίνηση του προγράμματος και περιλαμβάνει τον ορισμό:

του τρόπου λειτουργίας των χρονιστών (Timers).

μεταβλητών και σταθερών.

του τρόπου λειτουργίας του αισθητήρα.

των ανεξάρτητων συναρτήσεων που καλούνται από τον βρόγχο ελέγχου.

των εξωτερικών Interrupts.

των Pins που χρησιμοποιήθηκαν σαν εισόδους/εξόδους.

του πρωτοκόλλου επικοινωνίας I2C.

Στην συνέχεια στον βρόγχο ελέγχου πραγματοποιούνται με τη σειρά τα εξής:

Λήψη των ακατέργαστων δεδομένων από το αισθητήριο.

Υπολογισμός των γωνιακών ταχυτήτων ανά άξονα και των γωνιακών θέσεων ανά άξονα.

Παράλληλα με τον βρόγχο ελέγχου, με την χρήση των εξωτερικών interrupt, πραγματοποιείται λήψη των σημάτων από την τηλεκατεύθυνση.

Μετατροπή των σημάτων της τηλεκατεύθυνσης σε επιθυμητές γωνιακές θέσεις και συνολικής ώσης.

Εισαγωγή των δεδομένων στους ελεγκτές PI-D.

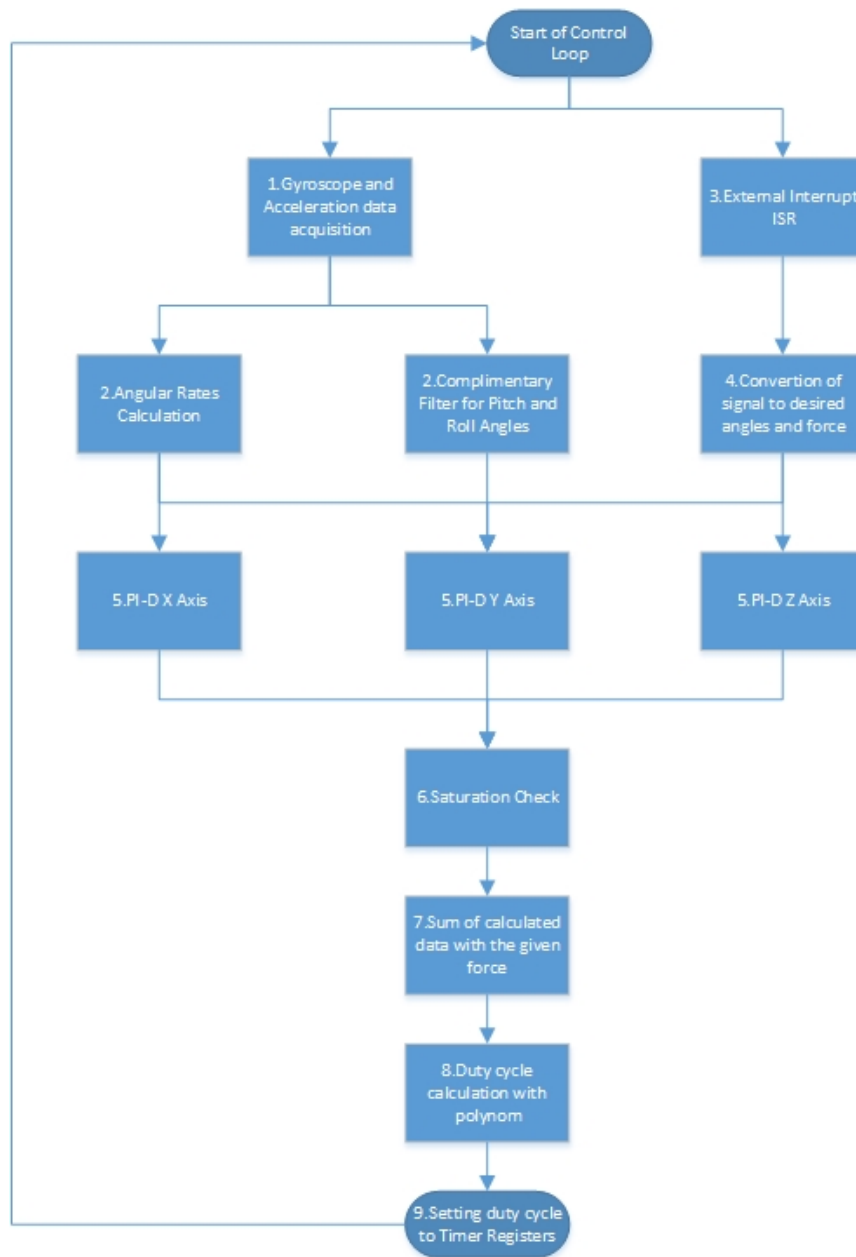
Έλεγχος για κορεσμό των τιμών από τους ελεγκτές, με όριο τα 150 γραμμάρια ώσης για τους άξονες X,Y και 50 γραμμάρια ώσης για τον άξονα Z.

Πρόσθεση των υπολογισμένων τιμών από τους ελεγκτές με την ώση που ζητείται από τον χρήστη.

Υπολογισμός του duty cycle του κάθε κινητήρα, με χρήση του αθροίσματος του προηγούμενου βήματος και των πολυωνύμων που προέκυψαν από την βαθμονόμηση.

Ορισμός των υπολογισμένων τιμών στους καταχωρητές των χρονιστών που αντιστοιχούν σε κάθε κινητήρα, για την παραγωγή του PWM σήματος.

Το διάγραμμα ροής του βρόγχου ελέγχου εικονίζεται στο επόμενο σχήμα:



Σχήμα 79 Διάγραμμα ροής βρόγχου ελέγχου

4 Σχόλια, Συμπεράσματα και Προτάσεις για Μελλοντική Εξέλιξη

4.1 Σχόλια, Συμπεράσματα

Συνολικά, η υπάμενη πλατφόρμα που αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής κατέδειξε την ανάγκη για περαιτέρω ανάπτυξη, διότι μόνο με τον αισθητήρα MPU6050 αυτό που καταλαβαίνει ο ελεγκτής είναι η στάση του στο χώρο, καθιστώντας την πτήση από κάποιον αρχάριο πολύ δύσκολη. Ένας σημαντικός παράγοντας στην πτήση της πλατφόρμας είναι το βάρος της, το οποίο μαζί με την χωρητικότητα της μπαταρίας, τους κινητήρες και τις έλικες καθορίζουν το χρόνο πτήσης. Επίσης λόγω της συμμετρικής σχεδίασης της κατασκευής δεν είναι εύκολο να ξεχωρίσει κανείς τον προσανατολισμό της πλατφόρμας, ειδικά σε μεγάλες αποστάσεις, με αποτέλεσμα να χάνεται ο προσανατολισμός που έχει ο χρήστης σε σχέση με τη πλατφόρμα και να παρέχονται επιθυμητές τιμές σε λάθος άξονα. Ο συνδυασμός βάρους, και συμμετρικής κατασκευής καθιστά μη δυνατή την προσθήκη αντικειμένων για σωστή αντίληψη της πλατφόρμας στο χώρο από τον χρήστη. Επίσης δεν εξαλείφει το συγκεκριμένο πρόβλημα στις μεγάλες αποστάσεις για παράδειγμα στα 30-40 μέτρα, όπου ο χρήστης δεν μπορεί να διακρίνει τα αντικείμενα πάνω στην πλατφόρμα. Με την προσθήκη ενός μαγνητόμετρου και σε συνδυασμό με τις μετρήσεις του γυροσκοπίου, οι μετρήσεις της γωνιακής θέσης θα είναι πλέον αξιόπιστες και ο ελεγκτής γωνιακής θέσης στον κατακόρυφο άξονα θα είναι υλοποιήσιμος. Επίσης σημαντικός παράγοντας στην πτήση μιας υπάμενης πλατφόρμας είναι ο καιρός, διότι μια δυνατή ριπή ανέμου μπορεί να αποβεί μοιραία, για την ακεραιότητα της πλατφόρμας, όπως ακόμα και για τον χρήστη ή κάποιον παρατηρητή.

4.2 Προτάσεις για Μελλοντική Εξέλιξη

Μια σειρά από βελτιώσεις έχουν προγραμματιστεί και εξετάζονται για την προσθήκη διάφορων αισθητηρίων. Τα αισθητήρια είναι τα εξής:

- Το βαρόμετρο είναι αισθητήριο μέτρησης ατμοσφαιρικής πίεσης σε συνδυασμό με την θερμοκρασία που επικρατεί μπορεί να υπολογιστεί το υψόμετρο που έχει επιτευχθεί
- Το σόναρ χρησιμοποιεί υπερήχους και με χρήση του χρόνου της αναπήδησης που γίνεται μεταξύ πλατφόρμας/εδάφους υπολογίζεται η μεταξύ τους απόσταση.

- Το μαγνητόμετρο είναι αισθητήριο μέτρησης των γωνιακών θέσεων στους 3 άξονες σε σχέση με την μαγνητική γραμμή που περνά από τον εκάστοτε τόπο.
- Το GPS είναι αισθητήριο που λαμβάνει δεδομένα μέσω ενός συνόλου δορυφόρων, με σκοπό τον υπολογισμό της γεωγραφικής θέσης, ταχύτητας, διεύθυνσης.

Με την βοήθεια τους η πλατφόρμα θα έχει περισσότερες λειτουργίες οι οποίες αναλύονται παρακάτω:

[A4] Λειτουργία διατήρησης ύψους, η οποία με τον συνδυασμό ενός βαρόμετρου και των μετρήσεων στον κατακόρυφο άξονα του επιταχυνσιόμετρου, επιτρέπει στον χρήστη να κινεί την πλατφόρμα προς όλες τις οριζόντιες κατευθύνσεις διατηρώντας το ίδιο ύψος.

[A5] Λειτουργία απογείωσης/προσγείωσης, η οποία με την χρήση ενός σόναρ μπορεί να απογειώσει/προσγειώσει όσο το δυνατόν πιο ομαλά την πλατφόρμα.

Λειτουργία κυκλικής τροχιάς, όπου με την χρήση GPS και τον υπολογισμό της επιθυμητής τροχιάς η πλατφόρμα κινείται σε κυκλική τροχιά γύρω από ένα αντικείμενο, κτίριο κτλ.

Λειτουργία Λόιτερ, όπου με την προσθήκη μαγνητόμετρου και σε συνδυασμό με το επιταχυνσιόμετρο και το βαρόμετρο η πλατφόρμα διατηρεί την θέση που έχει τόσο υψομετρικά όσο και στις υπόλοιπες κινήσεις.

Λειτουργία Άκρο, όπου ο χρήστης ελέγχει την γωνιακή ταχύτητα της πλατφόρμας ανά άξονα και όχι την γωνιακή θέση. Σημειώνεται πως σε αυτή τη λειτουργία πραγματοποιούνται επιτόπου αναστροφές στους άξονες X, Y και άλλες ακροβατικές κινήσεις.

Λειτουργία ακολούθησης χρήστη, όπου με τη χρήση μιας δεύτερης συσκευής GPS (κινητό/λάπτοπ κτλ.), η πλατφόρμα δέχεται εντολή να πετάξει στο σημείο που είναι η δεύτερη συσκευή GPS.

Λειτουργία επιστροφής στην αρχική θέση (βάση), όπου με τη χρήση GPS η πλατφόρμα μπορεί να επιστρέψει στην θέση που έχει οριστεί από το χρήστη ως θέση βάσης.

Βιβλιογραφία

Θα πρέπει να δίνεται ιδιαίτερη προσοχή στη σύνταξη της λίστας της βιβλιογραφίας, προκειμένου να αριθμούνται σωστά (βάσει της σειράς με την οποία εμφανίζονται στο κείμενο) και να υπάρχει ομοιομορφία στην σειρά παρουσίασης των στοιχείων της κάθε αναφοράς. Μελετήστε προσεκτικά τα παρακάτω υποδείγματα.

Άρθρα

- [A1] Réda Nouacer, Huascar Espinoza Ortiz, Yassine Ouhammou, Rodrigo Castiñeira González “*Framework of Key Enabling Technologies for Safe and Autonomous Drones' Applications*” 2019 22nd Euromicro Conference on Digital System Design (DSD).

- [A2] J. A.A. S. Somasiri, D. P. Chandima, A. G.B. P Jayasekara “*Extended Kalman Filter Based Autonomous Flying System for Quadcopters*” 2018 2nd International Conference On Electrical Engineering (EECon)

- [A3] Sarul Sakulthong, Suradet Tantrairatn, Watcharapol Saengphet “*Frequency response system identification and flight controller tuning for Quadcopter UAV*” 2018 Third International Conference on Engineering Science and Innovative Technology (ESIT)

- [A4] Zaki Mustapa, Shakir Saat, S. H. Husin, Norafizah Abas “*Altitude controller design for multi-copter UAV*” 2014 International Conference on Computer, Communications, and Control Technology (I4CT)

- [A5] Yohanes Gunawan Yusuf , Hendi Wicaksono Agung “*Optimizing ultrasonic and barometric sensors for quadcopter's altitude-hold using YoHe V1.2 PID and KK V2.0 boards*” 2016 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM)

Βιβλία και διδακτικά συγγράματα

- [B1] Μ. Σφακιωτάκης, «*Μικροελεγκτές*», Διδακτικές σημειώσεις, Τμήμα Ηλεκτρολογίας, Τ.Ε.Ι. Κρήτης.

- [B2] Μ. Σφακιωτάκης, «*Μικροελεγκτές*», Εργαστηριακές ασκήσεις, Τμήμα Ηλεκτρολογίας, Τ.Ε.Ι. Κρήτης.

- [B3] Ι. Μηναδάκης, «*Ηλεκτρονικές Κατασκευές*», Εργαστηριακές ασκήσεις, Τμήμα Ηλεκτρολογίας, Τ.Ε.Ι. Κρήτης.

- [B4] Μ. Σφακιωτάκης, «*Συστήματα Αυτομάτου Ελέγχου 2*», Διδακτικές σημειώσεις, Τμήμα Ηλεκτρολογίας, Τ.Ε.Ι. Κρήτης.

Ιστοσελίδες

- [I1] <http://shrediquette.blogspot.gr/p/shrediquette-dlx.html>
- [I2] <http://rcexplorer.se/product/tricopter-v3-kit/>
- [I3] <http://quadcopterdump.com/shop/drones/dji-phantom-2-vision/>
- [I4] <http://www.dronemakers.com/dji-inspire-1-tech-spec/>
- [I5] <https://www.wish.com/c/559642bd16ab354033f582f6>
- [I6] <http://shop.radiocontrolproducts.net/products/quadcopter-rtr-size-60-a>
- [I7] <http://store.xtremecoasters.com/products/spyder-extreme-fpv-hexacoaster>
- [I8] <http://www.myhelis.com/Spreading-Wings-S1000-Plus.html>
- [I9] <http://www.remoteaerialsurveys.co.uk/platforms/skyjib-8-titanium>
- [I10] https://en.wikipedia.org/wiki/Brushless_DC_electric_motor
- [I11] <http://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control>
- [I12] <http://www.helipal.com/skywalker-quattro-20a-x-4-speed-controller.html>
- [I13] <http://www.merq.com/files/Datasheet/emax.pdf>
- [I14] https://en.wikipedia.org/wiki/Lithium_polymer_battery
- [I15] <http://jkon.aeromodelling.gr/ninter-106.htm>
- [I16] <https://www.spektrumrc.com/Products/Default.aspx?ProdId=SPM6600>
- [I17] <http://arduino.cc/en/>
- [I18] <http://playground.arduino.cc/Main/MPU-6050>
- [I19] https://en.wikipedia.org/wiki/Inertial_measurement_unit

Τρικώπτερα

- [I1] <http://shrediquette.blogspot.gr/p/shrediquette-dlx.html>
- [I2] <http://rcexplorer.se/product/tricopter-v3-kit/>

Τετρακόπτερα

- [I3] <http://quadcopterdump.com/shop/drones/dji-phantom-2-vision/>
- [I4] <http://www.dronemakers.com/dji-inspire-1-tech-spec/>

Εξακόπτερα

- [I5] <https://www.wish.com/c/559642bd16ab354033f582f6>

[I6] <http://shop.radiocontrolproducts.net/products/quadcopter-rtr-size-60-a>

[I7] <http://store.xtremecoverters.com/products/spyder-extreme-fpv-hexacovert>

Οκτακόπιτερα

[I8] <http://www.myhelis.com/Spreading-Wings-S1000-Plus.html>

[I9] <http://www.remoteaerialsurveys.co.uk/platforms/skyjib-8-titanium>

Κεφάλαιο 2

[I10] https://en.wikipedia.org/wiki/Brushless_DC_electric_motor

[I11] <http://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control>

ESC :

[I12] <http://www.helipal.com/skywalker-quattro-20a-x-4-speed-controller.html>

Motor :

[I13] <http://www.merqc.com/files/Datasheet/emax.pdf>

Battery :

[I14] https://en.wikipedia.org/wiki/Lithium_polymer_battery

Propeller :

[I15] <http://jkon.aeromodelling.gr/ninter-106.htm>

Remote Control :

[I16] <https://www.spektrumrc.com/Products/Default.aspx?ProdId=SPM6600>

Arduino :

[I17] <http://arduino.cc/en/>

Mpu6050 :

[I18] <http://playground.arduino.cc/Main/MPU-6050>

IMU :

[I19] https://en.wikipedia.org/wiki/Inertial_measurement_unit

Παράρτημα-Α: Κώδικας Προγράμματος

```
// Ενσωμάτωση βιβλιοθηκών
#include <Wire.h> // Βιβλιοθήκη για το πρωτόκολλο I2C
#define MPU6050_AUX_VDDIO          0x01 // R/W
#define MPU6050_SMPLRT_DIV         0x00 // R/W
#define MPU6050_CONFIG              0x18 // R/W
#define MPU6050_GYRO_CONFIG        0x1B // R/W
#define MPU6050_ACCEL_CONFIG       0x1C // R/W
#define MPU6050_FF_THR              0x1D // R/W
#define MPU6050_FF_DUR              0x1E // R/W
#define MPU6050_MOT_THR             0x1F // R/W
#define MPU6050_MOT_DUR             0x20 // R/W
#define MPU6050_ZRMOT_THR          0x21 // R/W
#define MPU6050_ZRMOT_DUR          0x22 // R/W
#define MPU6050_FIFO_EN             0x23 // R/W
#define MPU6050_I2C_MST_CTRL       0x24 // R/W
#define MPU6050_I2C_SLV0_ADDR      0x25 // R/W
#define MPU6050_I2C_SLV0_REG       0x26 // R/W
#define MPU6050_I2C_SLV0_CTRL     0x27 // R/W
#define MPU6050_I2C_SLV1_ADDR      0x28 // R/W
#define MPU6050_I2C_SLV1_REG       0x29 // R/W
#define MPU6050_I2C_SLV1_CTRL     0x2A // R/W
#define MPU6050_I2C_SLV2_ADDR      0x2B // R/W
#define MPU6050_I2C_SLV2_REG       0x2C // R/W
#define MPU6050_I2C_SLV2_CTRL     0x2D // R/W
#define MPU6050_I2C_SLV3_ADDR      0x2E // R/W
#define MPU6050_I2C_SLV3_REG       0x2F // R/W
#define MPU6050_I2C_SLV3_CTRL     0x30 // R/W
#define MPU6050_I2C_SLV4_ADDR      0x31 // R/W
#define MPU6050_I2C_SLV4_REG       0x32 // R/W
#define MPU6050_I2C_SLV4_DO        0x33 // R/W
#define MPU6050_I2C_SLV4_CTRL     0x34 // R/W
#define MPU6050_I2C_SLV4_DI        0x35 // R
#define MPU6050_I2C_MST_STATUS     0x36 // R
#define MPU6050_INT_PIN_CFG        0x37 // R/W
#define MPU6050_INT_ENABLE         0x38 // R/W
#define MPU6050_INT_STATUS         0x3A // R
#define MPU6050_ACCEL_XOUT_H        0x3B // R
#define MPU6050_ACCEL_XOUT_L        0x3C // R
#define MPU6050_ACCEL_YOUT_H        0x3D // R
#define MPU6050_ACCEL_YOUT_L        0x3E // R
#define MPU6050_ACCEL_ZOUT_H        0x3F // R
#define MPU6050_ACCEL_ZOUT_L        0x40 // R
#define MPU6050_TEMP_OUT_H          0x41 // R
#define MPU6050_TEMP_OUT_L          0x42 // R
#define MPU6050_GYRO_XOUT_H         0x43 // R
```

```

#define MPU6050_GYRO_XOUT_L      0x44  // R
#define MPU6050_GYRO_YOUT_H      0x45  // R
#define MPU6050_GYRO_YOUT_L      0x46  // R
#define MPU6050_GYRO_ZOUT_H      0x47  // R
#define MPU6050_GYRO_ZOUT_L      0x48  // R
#define MPU6050_EXT_SENS_DATA_00  0x49  // R
#define MPU6050_EXT_SENS_DATA_01  0x4A  // R
#define MPU6050_EXT_SENS_DATA_02  0x4B  // R
#define MPU6050_EXT_SENS_DATA_03  0x4C  // R
#define MPU6050_EXT_SENS_DATA_04  0x4D  // R
#define MPU6050_EXT_SENS_DATA_05  0x4E  // R
#define MPU6050_EXT_SENS_DATA_06  0x4F  // R
#define MPU6050_EXT_SENS_DATA_07  0x50  // R
#define MPU6050_EXT_SENS_DATA_08  0x51  // R
#define MPU6050_EXT_SENS_DATA_09  0x52  // R
#define MPU6050_EXT_SENS_DATA_10  0x53  // R
#define MPU6050_EXT_SENS_DATA_11  0x54  // R
#define MPU6050_EXT_SENS_DATA_12  0x55  // R
#define MPU6050_EXT_SENS_DATA_13  0x56  // R
#define MPU6050_EXT_SENS_DATA_14  0x57  // R
#define MPU6050_EXT_SENS_DATA_15  0x58  // R
#define MPU6050_EXT_SENS_DATA_16  0x59  // R
#define MPU6050_EXT_SENS_DATA_17  0x5A  // R
#define MPU6050_EXT_SENS_DATA_18  0x5B  // R
#define MPU6050_EXT_SENS_DATA_19  0x5C  // R
#define MPU6050_EXT_SENS_DATA_20  0x5D  // R
#define MPU6050_EXT_SENS_DATA_21  0x5E  // R
#define MPU6050_EXT_SENS_DATA_22  0x5F  // R
#define MPU6050_EXT_SENS_DATA_23  0x60  // R
#define MPU6050_MOT_DETECT_STATUS  0x61  // R
#define MPU6050_I2C_SLV0_DO        0x63  // R/W
#define MPU6050_I2C_SLV1_DO        0x64  // R/W
#define MPU6050_I2C_SLV2_DO        0x65  // R/W
#define MPU6050_I2C_SLV3_DO        0x66  // R/W
#define MPU6050_I2C_MST_DELAY_CTRL  0x67  // R/W
#define MPU6050_SIGNAL_PATH_RESET  0x68  // R/W
#define MPU6050_MOT_DETECT_CTRL    0x69  // R/W
#define MPU6050_USER_CTRL          0x6A  // R/W
#define MPU6050_PWR_MGMT_1         0x6B  // R/W
#define MPU6050_PWR_MGMT_2         0x6C  // R/W
#define MPU6050_FIFO_COUNTH        0x72  // R/W
#define MPU6050_FIFO_COUNTL        0x73  // R/W
#define MPU6050_FIFO_R_W           0x74  // R/W
#define MPU6050_WHO_AM_I           0x75  // R

#define MPU6050_D0 0
#define MPU6050_D1 1
#define MPU6050_D2 2
#define MPU6050_D3 3
#define MPU6050_D4 4
#define MPU6050_D5 5

```

```

#define MPU6050_D6 6
#define MPU6050_D7 7

#define MPU6050_AUX_VDDIO MPU6050_D7

#define MPU6050_DLPF_CFG0 MPU6050_D0
#define MPU6050_DLPF_CFG1 MPU6050_D1
#define MPU6050_DLPF_CFG2 MPU6050_D2
#define MPU6050_EXT_SYNC_SET0 MPU6050_D3
#define MPU6050_EXT_SYNC_SET1 MPU6050_D4
#define MPU6050_EXT_SYNC_SET2 MPU6050_D5

#define MPU6050_EXT_SYNC_SET_0 (0)
#define MPU6050_EXT_SYNC_SET_1 (bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_2 (bit(MPU6050_EXT_SYNC_SET1))
#define MPU6050_EXT_SYNC_SET_3
(bit(MPU6050_EXT_SYNC_SET1)|bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_4 (bit(MPU6050_EXT_SYNC_SET2))
#define MPU6050_EXT_SYNC_SET_5
(bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_6
(bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET1))
#define MPU6050_EXT_SYNC_SET_7
(bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET1)|bit(MPU6050_EXT_SYNC
_SET0))

#define MPU6050_EXT_SYNC_DISABLED MPU6050_EXT_SYNC_SET_0
#define MPU6050_EXT_SYNC_TEMP_OUT_L MPU6050_EXT_SYNC_SET_1
#define MPU6050_EXT_SYNC_GYRO_XOUT_L MPU6050_EXT_SYNC_SET_2
#define MPU6050_EXT_SYNC_GYRO_YOUT_L MPU6050_EXT_SYNC_SET_3
#define MPU6050_EXT_SYNC_GYRO_ZOUT_L MPU6050_EXT_SYNC_SET_4
#define MPU6050_EXT_SYNC_ACCEL_XOUT_L MPU6050_EXT_SYNC_SET_5
#define MPU6050_EXT_SYNC_ACCEL_YOUT_L MPU6050_EXT_SYNC_SET_6
#define MPU6050_EXT_SYNC_ACCEL_ZOUT_L MPU6050_EXT_SYNC_SET_7

#define MPU6050_DLPF_CFG_0 (0)
#define MPU6050_DLPF_CFG_1 (bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_2 (bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_3 (bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_4 (bit(MPU6050_DLPF_CFG2))
#define MPU6050_DLPF_CFG_5 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_6 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_7
(bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))

#define MPU6050_DLPF_260HZ MPU6050_DLPF_CFG_0
#define MPU6050_DLPF_184HZ MPU6050_DLPF_CFG_1
#define MPU6050_DLPF_94HZ MPU6050_DLPF_CFG_2
#define MPU6050_DLPF_44HZ MPU6050_DLPF_CFG_3
#define MPU6050_DLPF_21HZ MPU6050_DLPF_CFG_4
#define MPU6050_DLPF_10HZ MPU6050_DLPF_CFG_5

```

```

#define MPU6050_DLPF_5HZ          MPU6050_DLPF_CFG_6
#define MPU6050_DLPF_RESERVED MPU6050_DLPF_CFG_7

#define MPU6050_FS_SEL0 MPU6050_D3
#define MPU6050_FS_SEL1 MPU6050_D4
#define MPU6050_ZG_ST    MPU6050_D5
#define MPU6050_YG_ST    MPU6050_D6
#define MPU6050_XG_ST    MPU6050_D7

#define MPU6050_FS_SEL_0 (0)
#define MPU6050_FS_SEL_1 (bit(MPU6050_FS_SEL0))
#define MPU6050_FS_SEL_2 (bit(MPU6050_FS_SEL1))
#define MPU6050_FS_SEL_3 (bit(MPU6050_FS_SEL1)|bit(MPU6050_FS_SEL0))

#define MPU6050_FS_SEL_250  MPU6050_FS_SEL_0
#define MPU6050_FS_SEL_500  MPU6050_FS_SEL_1
#define MPU6050_FS_SEL_1000 MPU6050_FS_SEL_2
#define MPU6050_FS_SEL_2000 MPU6050_FS_SEL_3

#define MPU6050_ACCEL_HPF0 MPU6050_D0
#define MPU6050_ACCEL_HPF1 MPU6050_D1
#define MPU6050_ACCEL_HPF2 MPU6050_D2
#define MPU6050_AFS_SEL0   MPU6050_D3
#define MPU6050_AFS_SEL1   MPU6050_D4
#define MPU6050_ZA_ST      MPU6050_D5
#define MPU6050_YA_ST      MPU6050_D6
#define MPU6050_XA_ST      MPU6050_D7

#define MPU6050_ACCEL_HPF_0 (0)
#define MPU6050_ACCEL_HPF_1 (bit(MPU6050_ACCEL_HPF0))
#define MPU6050_ACCEL_HPF_2 (bit(MPU6050_ACCEL_HPF1))
#define MPU6050_ACCEL_HPF_3
(bit(MPU6050_ACCEL_HPF1)|bit(MPU6050_ACCEL_HPF0))
#define MPU6050_ACCEL_HPF_4 (bit(MPU6050_ACCEL_HPF2))
#define MPU6050_ACCEL_HPF_7
(bit(MPU6050_ACCEL_HPF2)|bit(MPU6050_ACCEL_HPF1)|bit(MPU6050_ACCEL_HPF0))

#define MPU6050_ACCEL_HPF_RESET MPU6050_ACCEL_HPF_0
#define MPU6050_ACCEL_HPF_5HZ   MPU6050_ACCEL_HPF_1
#define MPU6050_ACCEL_HPF_2_5HZ MPU6050_ACCEL_HPF_2
#define MPU6050_ACCEL_HPF_1_25HZ MPU6050_ACCEL_HPF_3
#define MPU6050_ACCEL_HPF_0_63HZ MPU6050_ACCEL_HPF_4
#define MPU6050_ACCEL_HPF_HOLD  MPU6050_ACCEL_HPF_7

#define MPU6050_AFS_SEL_0 (0)
#define MPU6050_AFS_SEL_1 (bit(MPU6050_AFS_SEL0))
#define MPU6050_AFS_SEL_2 (bit(MPU6050_AFS_SEL1))
#define MPU6050_AFS_SEL_3 (bit(MPU6050_AFS_SEL1)|bit(MPU6050_AFS_SEL0))

#define MPU6050_AFS_SEL_2G MPU6050_AFS_SEL_0
#define MPU6050_AFS_SEL_4G MPU6050_AFS_SEL_1

```

```

#define MPU6050_AFS_SEL_8G MPU6050_AFS_SEL_2
#define MPU6050_AFS_SEL_16G MPU6050_AFS_SEL_3

#define MPU6050_SLV0_FIFO_EN MPU6050_D0
#define MPU6050_SLV1_FIFO_EN MPU6050_D1
#define MPU6050_SLV2_FIFO_EN MPU6050_D2
#define MPU6050_ACCEL_FIFO_EN MPU6050_D3
#define MPU6050_ZG_FIFO_EN MPU6050_D4
#define MPU6050_YG_FIFO_EN MPU6050_D5
#define MPU6050_XG_FIFO_EN MPU6050_D6
#define MPU6050_TEMP_FIFO_EN MPU6050_D7

#define MPU6050_I2C_MST_CLK0 MPU6050_D0
#define MPU6050_I2C_MST_CLK1 MPU6050_D1
#define MPU6050_I2C_MST_CLK2 MPU6050_D2
#define MPU6050_I2C_MST_CLK3 MPU6050_D3
#define MPU6050_I2C_MST_P_NSR MPU6050_D4
#define MPU6050_SLV_3_FIFO_EN MPU6050_D5
#define MPU6050_WAIT_FOR_ES MPU6050_D6
#define MPU6050_MULT_MST_EN MPU6050_D7

#define MPU6050_I2C_MST_CLK_0 (0)
#define MPU6050_I2C_MST_CLK_1 (bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_2 (bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_3
(bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_4 (bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_5
(bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_6
(bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_7
(bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_8 (bit(MPU6050_I2C_MST_CLK3))
#define MPU6050_I2C_MST_CLK_9
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_10
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_11
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_12
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_13
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_14
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))

```

```

#define MPU6050_I2C_MST_CLK_15
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))

#define MPU6050_I2C_MST_CLK_348KHZ MPU6050_I2C_MST_CLK_0
#define MPU6050_I2C_MST_CLK_333KHZ MPU6050_I2C_MST_CLK_1
#define MPU6050_I2C_MST_CLK_320KHZ MPU6050_I2C_MST_CLK_2
#define MPU6050_I2C_MST_CLK_308KHZ MPU6050_I2C_MST_CLK_3
#define MPU6050_I2C_MST_CLK_296KHZ MPU6050_I2C_MST_CLK_4
#define MPU6050_I2C_MST_CLK_286KHZ MPU6050_I2C_MST_CLK_5
#define MPU6050_I2C_MST_CLK_276KHZ MPU6050_I2C_MST_CLK_6
#define MPU6050_I2C_MST_CLK_267KHZ MPU6050_I2C_MST_CLK_7
#define MPU6050_I2C_MST_CLK_258KHZ MPU6050_I2C_MST_CLK_8
#define MPU6050_I2C_MST_CLK_500KHZ MPU6050_I2C_MST_CLK_9
#define MPU6050_I2C_MST_CLK_471KHZ MPU6050_I2C_MST_CLK_10
#define MPU6050_I2C_MST_CLK_444KHZ MPU6050_I2C_MST_CLK_11
#define MPU6050_I2C_MST_CLK_421KHZ MPU6050_I2C_MST_CLK_12
#define MPU6050_I2C_MST_CLK_400KHZ MPU6050_I2C_MST_CLK_13
#define MPU6050_I2C_MST_CLK_381KHZ MPU6050_I2C_MST_CLK_14
#define MPU6050_I2C_MST_CLK_364KHZ MPU6050_I2C_MST_CLK_15

#define MPU6050_I2C_SLV0_RW MPU6050_D7

#define MPU6050_I2C_SLV0_LEN0 MPU6050_D0
#define MPU6050_I2C_SLV0_LEN1 MPU6050_D1
#define MPU6050_I2C_SLV0_LEN2 MPU6050_D2
#define MPU6050_I2C_SLV0_LEN3 MPU6050_D3
#define MPU6050_I2C_SLV0_GRP MPU6050_D4
#define MPU6050_I2C_SLV0_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV0_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV0_EN MPU6050_D7

#define MPU6050_I2C_SLV0_LEN_MASK 0x0F

#define MPU6050_I2C_SLV1_RW MPU6050_D7

#define MPU6050_I2C_SLV1_LEN0 MPU6050_D0
#define MPU6050_I2C_SLV1_LEN1 MPU6050_D1
#define MPU6050_I2C_SLV1_LEN2 MPU6050_D2
#define MPU6050_I2C_SLV1_LEN3 MPU6050_D3
#define MPU6050_I2C_SLV1_GRP MPU6050_D4
#define MPU6050_I2C_SLV1_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV1_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV1_EN MPU6050_D7

#define MPU6050_I2C_SLV1_LEN_MASK 0x0F

#define MPU6050_I2C_SLV2_RW MPU6050_D7

#define MPU6050_I2C_SLV2_LEN0 MPU6050_D0
#define MPU6050_I2C_SLV2_LEN1 MPU6050_D1

```

```

#define MPU6050_I2C_SLV2_LEN2      MPU6050_D2
#define MPU6050_I2C_SLV2_LEN3      MPU6050_D3
#define MPU6050_I2C_SLV2_GRP       MPU6050_D4
#define MPU6050_I2C_SLV2_REG_DIS   MPU6050_D5
#define MPU6050_I2C_SLV2_BYTE_SW   MPU6050_D6
#define MPU6050_I2C_SLV2_EN        MPU6050_D7

#define MPU6050_I2C_SLV2_LEN_MASK  0x0F

#define MPU6050_I2C_SLV3_RW        MPU6050_D7

#define MPU6050_I2C_SLV3_LEN0      MPU6050_D0
#define MPU6050_I2C_SLV3_LEN1      MPU6050_D1
#define MPU6050_I2C_SLV3_LEN2      MPU6050_D2
#define MPU6050_I2C_SLV3_LEN3      MPU6050_D3
#define MPU6050_I2C_SLV3_GRP       MPU6050_D4
#define MPU6050_I2C_SLV3_REG_DIS   MPU6050_D5
#define MPU6050_I2C_SLV3_BYTE_SW   MPU6050_D6
#define MPU6050_I2C_SLV3_EN        MPU6050_D7

#define MPU6050_I2C_SLV3_LEN_MASK  0x0F

#define MPU6050_I2C_SLV4_RW        MPU6050_D7

#define MPU6050_I2C_MST_DLY0       MPU6050_D0
#define MPU6050_I2C_MST_DLY1       MPU6050_D1
#define MPU6050_I2C_MST_DLY2       MPU6050_D2
#define MPU6050_I2C_MST_DLY3       MPU6050_D3
#define MPU6050_I2C_MST_DLY4       MPU6050_D4
#define MPU6050_I2C_SLV4_REG_DIS   MPU6050_D5
#define MPU6050_I2C_SLV4_INT_EN    MPU6050_D6
#define MPU6050_I2C_SLV4_EN        MPU6050_D7

#define MPU6050_I2C_MST_DLY_MASK    0x1F

#define MPU6050_I2C_SLV0_NACK      MPU6050_D0
#define MPU6050_I2C_SLV1_NACK      MPU6050_D1
#define MPU6050_I2C_SLV2_NACK      MPU6050_D2
#define MPU6050_I2C_SLV3_NACK      MPU6050_D3
#define MPU6050_I2C_SLV4_NACK      MPU6050_D4
#define MPU6050_I2C_LOST_ARB       MPU6050_D5
#define MPU6050_I2C_SLV4_DONE      MPU6050_D6
#define MPU6050_PASS_THROUGH       MPU6050_D7

#define MPU6050_CLKOUT_EN           MPU6050_D0
#define MPU6050_I2C_BYPASS_EN       MPU6050_D1
#define MPU6050_FSYNC_INT_EN        MPU6050_D2
#define MPU6050_FSYNC_INT_LEVEL    MPU6050_D3
#define MPU6050_INT_RD_CLEAR        MPU6050_D4
#define MPU6050_LATCH_INT_EN        MPU6050_D5
#define MPU6050_INT_OPEN            MPU6050_D6

```

```

#define MPU6050_INT_LEVEL      MPU6050_D7

#define MPU6050_DATA_RDY_EN    MPU6050_D0
#define MPU6050_I2C_MST_INT_EN MPU6050_D3
#define MPU6050_FIFO_OFLOW_EN  MPU6050_D4
#define MPU6050_ZMOT_EN        MPU6050_D5
#define MPU6050_MOT_EN         MPU6050_D6
#define MPU6050_FF_EN          MPU6050_D7

#define MPU6050_DATA_RDY_INT   MPU6050_D0
#define MPU6050_I2C_MST_INT    MPU6050_D3
#define MPU6050_FIFO_OFLOW_INT MPU6050_D4
#define MPU6050_ZMOT_INT       MPU6050_D5
#define MPU6050_MOT_INT        MPU6050_D6
#define MPU6050_FF_INT         MPU6050_D7

#define MPU6050_MOT_ZRMOT      MPU6050_D0
#define MPU6050_MOT_ZPOS       MPU6050_D2
#define MPU6050_MOT_ZNEG       MPU6050_D3
#define MPU6050_MOT_YPOS       MPU6050_D4
#define MPU6050_MOT_YNEG       MPU6050_D5
#define MPU6050_MOT_XPOS       MPU6050_D6
#define MPU6050_MOT_XNEG       MPU6050_D7

#define MPU6050_I2C_SLV0_DLY_EN MPU6050_D0
#define MPU6050_I2C_SLV1_DLY_EN MPU6050_D1
#define MPU6050_I2C_SLV2_DLY_EN MPU6050_D2
#define MPU6050_I2C_SLV3_DLY_EN MPU6050_D3
#define MPU6050_I2C_SLV4_DLY_EN MPU6050_D4
#define MPU6050_DELAY_ES_SHADOW MPU6050_D7

#define MPU6050_TEMP_RESET     MPU6050_D0
#define MPU6050_ACCEL_RESET    MPU6050_D1
#define MPU6050_GYRO_RESET     MPU6050_D2

#define MPU6050_MOT_COUNT0     MPU6050_D0
#define MPU6050_MOT_COUNT1     MPU6050_D1
#define MPU6050_FF_COUNT0      MPU6050_D2
#define MPU6050_FF_COUNT1      MPU6050_D3
#define MPU6050_ACCEL_ON_DELAY0 MPU6050_D4
#define MPU6050_ACCEL_ON_DELAY1 MPU6050_D5

#define MPU6050_MOT_COUNT_0 (0)
#define MPU6050_MOT_COUNT_1 (bit(MPU6050_MOT_COUNT0))
#define MPU6050_MOT_COUNT_2 (bit(MPU6050_MOT_COUNT1))
#define MPU6050_MOT_COUNT_3
(bit(MPU6050_MOT_COUNT1)|bit(MPU6050_MOT_COUNT0))

#define MPU6050_MOT_COUNT_RESET MPU6050_MOT_COUNT_0

#define MPU6050_FF_COUNT_0 (0)

```



```

#define MPU6050_FF_COUNT_1 (bit(MPU6050_FF_COUNT0))
#define MPU6050_FF_COUNT_2 (bit(MPU6050_FF_COUNT1))
#define MPU6050_FF_COUNT_3 (bit(MPU6050_FF_COUNT1)|bit(MPU6050_FF_COUNT0))

#define MPU6050_FF_COUNT_RESET MPU6050_FF_COUNT_0

#define MPU6050_ACCEL_ON_DELAY_0 (0)
#define MPU6050_ACCEL_ON_DELAY_1 (bit(MPU6050_ACCEL_ON_DELAY0))
#define MPU6050_ACCEL_ON_DELAY_2 (bit(MPU6050_ACCEL_ON_DELAY1))
#define MPU6050_ACCEL_ON_DELAY_3
(bit(MPU6050_ACCEL_ON_DELAY1)|bit(MPU6050_ACCEL_ON_DELAY0))

#define MPU6050_ACCEL_ON_DELAY_0MS MPU6050_ACCEL_ON_DELAY_0
#define MPU6050_ACCEL_ON_DELAY_1MS MPU6050_ACCEL_ON_DELAY_1
#define MPU6050_ACCEL_ON_DELAY_2MS MPU6050_ACCEL_ON_DELAY_2
#define MPU6050_ACCEL_ON_DELAY_3MS MPU6050_ACCEL_ON_DELAY_3

#define MPU6050_SIG_COND_RESET MPU6050_D0
#define MPU6050_I2C_MST_RESET MPU6050_D1
#define MPU6050_FIFO_RESET MPU6050_D2
#define MPU6050_I2C_IF_DIS MPU6050_D4 // Πρέπει να είναι 0 για τον
MPU-6050
#define MPU6050_I2C_MST_EN MPU6050_D5
#define MPU6050_FIFO_EN MPU6050_D6

#define MPU6050_CLKSEL0 MPU6050_D0
#define MPU6050_CLKSEL1 MPU6050_D1
#define MPU6050_CLKSEL2 MPU6050_D2
#define MPU6050_TEMP_DIS MPU6050_D3 // 1: Αδρανοποίηση αισθητήρα
θερμοκρασίας
#define MPU6050_CYCLE MPU6050_D5 // 1: Δειγματοληψία και
αδρανοποίηση
#define MPU6050_SLEEP MPU6050_D6 // 1: Λειτουργία αδράνειας
#define MPU6050_DEVICE_RESET MPU6050_D7 // 1: Επαναφορά προεπιλογών

#define MPU6050_CLKSEL_0 (0)
#define MPU6050_CLKSEL_1 (bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_2 (bit(MPU6050_CLKSEL1))
#define MPU6050_CLKSEL_3 (bit(MPU6050_CLKSEL1)|bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_4 (bit(MPU6050_CLKSEL2))
#define MPU6050_CLKSEL_5 (bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_6 (bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL1))
#define MPU6050_CLKSEL_7
(bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL1)|bit(MPU6050_CLKSEL0))

#define MPU6050_CLKSEL_INTERNAL MPU6050_CLKSEL_0
#define MPU6050_CLKSEL_X MPU6050_CLKSEL_1
#define MPU6050_CLKSEL_Y MPU6050_CLKSEL_2
#define MPU6050_CLKSEL_Z MPU6050_CLKSEL_3
#define MPU6050_CLKSEL_EXT_32KHZ MPU6050_CLKSEL_4
#define MPU6050_CLKSEL_EXT_19_2MHZ MPU6050_CLKSEL_5

```

```

#define MPU6050_CLKSEL_RESERVED    MPU6050_CLKSEL_6
#define MPU6050_CLKSEL_STOP        MPU6050_CLKSEL_7

#define MPU6050_STBY_ZG            MPU6050_D0
#define MPU6050_STBY_YG            MPU6050_D1
#define MPU6050_STBY_XG            MPU6050_D2
#define MPU6050_STBY_ZA            MPU6050_D3
#define MPU6050_STBY_YA            MPU6050_D4
#define MPU6050_STBY_XA            MPU6050_D5
#define MPU6050_LP_WAKE_CTRL0      MPU6050_D6
#define MPU6050_LP_WAKE_CTRL1      MPU6050_D7

#define MPU6050_LP_WAKE_CTRL_0 (0)
#define MPU6050_LP_WAKE_CTRL_1 (bit(MPU6050_LP_WAKE_CTRL0))
#define MPU6050_LP_WAKE_CTRL_2 (bit(MPU6050_LP_WAKE_CTRL1))
#define MPU6050_LP_WAKE_CTRL_3
(bit(MPU6050_LP_WAKE_CTRL1)|bit(MPU6050_LP_WAKE_CTRL0))

#define MPU6050_LP_WAKE_1_25HZ     MPU6050_LP_WAKE_CTRL_0
#define MPU6050_LP_WAKE_2_5HZ      MPU6050_LP_WAKE_CTRL_1
#define MPU6050_LP_WAKE_5HZ        MPU6050_LP_WAKE_CTRL_2
#define MPU6050_LP_WAKE_10HZ       MPU6050_LP_WAKE_CTRL_3

#define MPU6050_I2C_ADDRESS 0x68
typedef union accel_t_gyro_union
{
    struct
    {
        uint8_t x_accel_h;
        uint8_t x_accel_l;
        uint8_t y_accel_h;
        uint8_t y_accel_l;
        uint8_t z_accel_h;
        uint8_t z_accel_l;
        uint8_t t_h;
        uint8_t t_l;
        uint8_t x_gyro_h;
        uint8_t x_gyro_l;
        uint8_t y_gyro_h;
        uint8_t y_gyro_l;
        uint8_t z_gyro_h;
        uint8_t z_gyro_l;
    } reg;
    struct
    {
        int x_accel;
        int y_accel;
        int z_accel;
        int temperature;
    }
};

```

```

    int x_gyro;
    int y_gyro;
    int z_gyro;
} value;
};

// Χρήση των παρακάτω παγκόσμιων μεταβλητών για αποθήκευση
// της περιστροφής γωνίας του αισθητήρα
unsigned long last_read_time;
// Αυτές είναι οι φιλτραρισμένες γωνίες
float      last_x_angle;
float      last_y_angle;
float      last_z_angle;
// γωνίες του γυροσκοπίου για την σύγκριση της μετάπτωσης
float      last_gyro_x_angle;
float      last_gyro_y_angle;
float      last_gyro_z_angle;

void set_last_read_angle_data(unsigned long time, float x, float y, float
z, float x_gyro, float y_gyro, float z_gyro) {
    last_read_time = time;
    last_x_angle = x;
    last_y_angle = y;
    last_z_angle = z;
    last_gyro_x_angle = x_gyro;
    last_gyro_y_angle = y_gyro;
    last_gyro_z_angle = z_gyro;
}

inline unsigned long get_last_time() {return last_read_time;}
inline float get_last_x_angle() {return last_x_angle;}
inline float get_last_y_angle() {return last_y_angle;}
inline float get_last_z_angle() {return last_z_angle;}
inline float get_last_gyro_x_angle() {return last_gyro_x_angle;}
inline float get_last_gyro_y_angle() {return last_gyro_y_angle;}
inline float get_last_gyro_z_angle() {return last_gyro_z_angle;}

// Χρήση των επόμενων μεταβλητών παγκοσμίου επιπέδου για πρόσβαση στις
// συναρτήσεις για την βαθμονόμηση του επιταχυνσιομέτρου
float      base_x_accel;
float      base_y_accel;
float      base_z_accel;

float      base_x_gyro;
float      base_y_gyro;
float      base_z_gyro;

int read_gyro_accel_vals(uint8_t* accel_t_gyro_ptr) {
    // Ανάγνωση των ανεπεξέργαστων τιμών

```

```

// Ανάγνωση 14 bytes με την μία εμπεριέχοντας επιτάχυνση, θερμοκρασία και
γωνιακή ταχύτητα.
accel_t_gyro_union* accel_t_gyro = (accel_t_gyro_union *)
accel_t_gyro_ptr;

int error = MPU6050_read (MPU6050_ACCEL_XOUT_H, (uint8_t *) accel_t_gyro,
sizeof(*accel_t_gyro));

// Εναλλαγή όλων των high και low bytes.
// Μετα από αυτό όλες οι τιμές των καταψωρητών εναλλάσσονται έτσι ώστε η
δομή με το όνομα px_x_accel_l να μην περιέχει το lower byte.
uint8_t swap;
#define SWAP(x,y) swap = x; x = y; y = swap

SWAP ((*accel_t_gyro).reg.x_accel_h, (*accel_t_gyro).reg.x_accel_l);
SWAP ((*accel_t_gyro).reg.y_accel_h, (*accel_t_gyro).reg.y_accel_l);
SWAP ((*accel_t_gyro).reg.z_accel_h, (*accel_t_gyro).reg.z_accel_l);
SWAP ((*accel_t_gyro).reg.t_h, (*accel_t_gyro).reg.t_l);
SWAP ((*accel_t_gyro).reg.x_gyro_h, (*accel_t_gyro).reg.x_gyro_l);
SWAP ((*accel_t_gyro).reg.y_gyro_h, (*accel_t_gyro).reg.y_gyro_l);
SWAP ((*accel_t_gyro).reg.z_gyro_h, (*accel_t_gyro).reg.z_gyro_l);

return error;
}

// Συνάρτηση βαθμονόμησης αισθητήρα
void calibrate_sensors() {
int num_readings = 10;
float x_accel = 0;
float y_accel = 0;
float z_accel = 0;
float x_gyro = 0;
float y_gyro = 0;
float z_gyro = 0;
accel_t_gyro_union accel_t_gyro;

// Απόρριψη των πρώτων μετρήσεων του αισθητήρα
read_gyro_accel_vals((uint8_t *) &accel_t_gyro);

// Ανάγνωση των δεδομένων από τον αισθητήρα και υπολογισμός του μέσου
όρου τους
for (int i = 0; i < num_readings; i++) {
read_gyro_accel_vals((uint8_t *) &accel_t_gyro);
x_accel += accel_t_gyro.value.x_accel;
y_accel += accel_t_gyro.value.y_accel;
z_accel += accel_t_gyro.value.z_accel;
x_gyro += accel_t_gyro.value.x_gyro;
y_gyro += accel_t_gyro.value.y_gyro;
z_gyro += accel_t_gyro.value.z_gyro;
delay(100);
}

```

```

}
x_accel /= num_readings;
y_accel /= num_readings;
z_accel /= num_readings;
x_gyro /= num_readings;
y_gyro /= num_readings;
z_gyro /= num_readings;

// Αποθήκευση των ανεπεξέργαστων τιμών από την βαθμονόμηση σε παγκόσμιο
επίπεδο
base_x_accel = x_accel;
base_y_accel = y_accel;
base_z_accel = z_accel;
base_x_gyro = x_gyro;
base_y_gyro = y_gyro;
base_z_gyro = z_gyro;

}
// Μεταβλητές για συνάρτηση του Throttle Channel με αντιστοιχία στο pin 2
unsigned long Rc_Thr_flag1=0;
unsigned long Rc_Thr_flag2=0;
unsigned long Rc_Throttle_ch=0;
int Throttle_counter=0;

// Μεταβλητές για συνάρτηση του Roll Channel με αντιστοιχία στο pin 2
unsigned long Rc_Roll_flag1=0;
unsigned long Rc_Roll_flag2=0;
unsigned long Rc_Roll_ch=0;
int Roll_counter=0;

// Μεταβλητές για συνάρτηση του Pitch Channel με αντιστοιχία στο pin 2
unsigned long Rc_Pitch_flag1=0;
unsigned long Rc_Pitch_flag2=0;
unsigned long Rc_Pitch_ch=0;
int Pitch_counter=0;

#define DLAY_us 4000 //σταθερά για την συχνότητα του βρόχου ελέγχου
// Ορισμός των pins για κάθε PWM
#define pwmPin0 13 //timer 1 OCR1C ΚΙΝΗΤΗΡΑΣ 2
#define pwmPin1 12 //timer 1 OCR1B ΚΙΝΗΤΗΡΑΣ 1
#define pwmPin2 11 //timer 1 OCR1A ΚΙΝΗΤΗΡΑΣ 4
#define pwmPin3 6 //timer 4 OCR4A ΚΙΝΗΤΗΡΑΣ 3

// Σταθερά του χρόνου του βρόχου ελέγχου
#define T 0.002
volatile char wait=0;
int P=0;
// Ορισμός της μεταβλητής της συνολικής δύναμης
float force=0;
// Ορισμός των σταθερών KP για κάθε άξονα (x,y,z)

```

```

float Kpy=2.8;
float Kpx=2.8;
float Kpz=10;

// Ορισμός των σταθερών KD για κάθε άξονα (x,y,z)
float Kdx=1;
float Kdy=1;
float Kdz=2;

// Ορισμός των σταθερών KI για κάθε άξονα (x,y,z)
float Kix=2.5;
float Kiy=2.5;
float Kiz=2.5;

// Σήματα (x,y,z)
float phi_signal=0;
float theta_signal=0;
float psi_signal=0;

// Μεταβλητές για τις επιθυμητές γωνιακές ταχύτητες ανά άξονα (x,y,z)
float desx=0;
float desy=0;
float desz=0;

// Μεταβλητές για τις επιθυμητές γωνίες ανά άξονα (x,y,z)
float phi=0;
float theta=0;
float psi=0;

// Μεταβλητές σφαλμάτων ανά άξονα (x,y,z)
float phi_error=0;
float theta_error=0;
float psi_error=0;

// Μεταβλητές σημάτων γωνιακής ταχύτητας ανά άξονα (x,y,z)
float rate_signalx=0;
float rate_signaly=0;
float rate_signalz=0;

// Σφάλματα γωνιακής ταχύτητας ανά άξονα (x,y,z)
float rate_errorx=0;
float rate_errory=0;
float rate_errorz=0;

// Μεταβλητές εξόδων των ελεγκτών ανά άξονα (x,y,z)
float rate_x=0;
float rate_y=0;
float rate_z=0;

// Μεταβλητές για τις σταθερές των πολυονύμων ανά κινητήρα
float m3_x3 =(3.1057/1000000000);

```

```

float m3_x2=(-8.427/100000);
float m3_x1=(0.1485);
float m3_x0=(0.0969);

float m8_x3 =(1.5288/10000000);
float m8_x2=(-1.8153/10000);
float m8_x1=(0.1585);
float m8_x0=(-0.2505);

float m7_x3 =(3.6572/100000000);
float m7_x2=(-7.5659/100000);
float m7_x1=(0.1416);
float m7_x0=(0.0070);

float m6_x3 =(7.23/100000000);
float m6_x2=(-1.1694/10000);
float m6_x1=(0.1503);
float m6_x0=(0.3691);

// Μεταβλητές που χρησιμοποιούνται σαν σημαίες κορεσμού
int saturated_x=0;
int saturated_y=0;
int saturated_z=0;

// Μεταβήτες ολοκλήρωσης ανά άξονα (x,y,z)
float Integral_x=0;
float Integral_y=0;
float Integral_z=0;

////////// ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ //////////

void setup()
{

// Ορισμός του εξωτερικού interrupt στο pin 4 σε λειτουργία change με
συνάρτηση την pitch_isr
attachInterrupt(4, Pitch_isr, CHANGE);

// Ορισμός του εξωτερικού interrupt στο pin 1 σε λειτουργία change με
συνάρτηση την throttle_isr
attachInterrupt(1, Throttle_isr, CHANGE);

// Ορισμός του εξωτερικού interrupt στο pin 5 σε λειτουργία change με
συνάρτηση την roll_isr
attachInterrupt(5, Roll_isr, CHANGE);
int error;
uint8_t c;

// Καθορισμός των pins των PWM σε εξόδους
pinMode(pwmPin0,OUTPUT);
pinMode(pwmPin1,OUTPUT);

```

```

pinMode (pwmPin2, OUTPUT);
pinMode (pwmPin3, OUTPUT);

// Σειάρισμα ολόκληρου του καταχωρητή TCCR1A σε 0 για CTC mode
TCCR5A=0b00000000;

// Το ίδιο για τον καταχωρητή TCCR1B
TCCR5B=0b00001010;

// Αρχικοποίηση τιμής του μετρητή στο 0
TCNT5=0;
OCR5A = DLAY_us;

// Καταχωρητής ορισμού λειτουργίας interrupt του Timer 5
TIMSK5 = 0x02;

// Έχουμε clock 16MHz άρα για 8 prescaler έχουμε μοναδιαία απαρίθμηση
0,5μsec
// Ara gia na piasoume ta 2 msec theloume 2000/0,5=4000 vhmata
// CTC mode and prescaler 8
// Καθορισμός του timer 1 σε λειτουργία phase correct με prescaler 8 για
παραγωγή PWM με άνω όριο τον καταχωρητή ICR1 με τιμή 22000

// Ορισμός λειτουργίας των καταχωρητών OCR1A, OCR1B, OCR1C σε HIGH όταν ο
μετρητής φτάσει στο άνω όριο και σε LOW όταν ο μετρητής φτάσει στο 0
TCCR1A=0b11111110;
TCCR1B=0b00010010;
// Καταχωρητής που χρησιμοποιείται ως άνω όριο του μετρητή
ICR1=22000;

// Καθορισμός του timer 4 σε λειτουργία phase correct με prescaler 8 για
παραγωγή PWM με άνω όριο τον καταχωρητή ICR1 με τιμή 22000
// Ορισμός λειτουργίας των καταχωρητών OCR1A, OCR1B, OCR1C σε HIGH όταν ο
μετρητής φτάσει στο άνω όριο και σε LOW όταν ο μετρητής φτάσει στο 0
// Έχουμε clock 16MHz άρα για 8 prescaler έχουμε μοναδιαία απαρίθμηση
0,5μsec
TCCR4A=0b11111110;
TCCR4B=0b00010010;
ICR4=22000;

OCR1A=21000; // Ορισμός του PWM στα 2.1 msec για αρχικοποίηση του Esc
OCR1B=21000; // Ορισμός του PWM στα 2.1 msec για αρχικοποίηση του Esc
OCR1C=21000; // Ορισμός του PWM στα 2.1 msec για αρχικοποίηση του Esc
OCR4A=21000; // Ορισμός του PWM στα 2.1 msec για αρχικοποίηση του Esc
delay(5000); // Αναμονή 5 δευτερολέπτων

OCR1A=20900; // Ορισμός του PWM στην ελάχιστη θέση όπου ο κινητήρας δεν
κινείται
OCR1B=20900; // Ορισμός του PWM στην ελάχιστη θέση όπου ο κινητήρας δεν
κινείται

```



```

OCR1C=20900; // Ορισμός του PWM στην ελάχιστη θέση όπου ο κινητήρας δεν
κινείται
OCR4A=20900; // Ορισμός του PWM στην ελάχιστη θέση όπου ο κινητήρας δεν
κινείται

// Εκκίνηση της σειριακής επικοινωνίας σε 115200 Baudrate
Serial.begin(115200);
Wire.begin(); // Εκκίνηση του πρωτοκόλου I2C
TWBR = 1; // Ορισμός της συχνότητας του πρωτοκόλου I2C

error = MPU6050_read (MPU6050_WHO_AM_I, &c, 1);

error = MPU6050_read (MPU6050_PWR_MGMT_2, &c, 1);

MPU6050_write_reg (MPU6050_PWR_MGMT_1, 0);

// Αρχικοποίηση των γωνιών
calibrate_sensors();
set_last_read_angle_data(millis(), 0, 0, 0, 0, 0, 0);
}

//////////////////// ΒΡΟΧΟΣ ΕΛΕΓΧΟΥ////////////////////////////////////
void loop()
{
  wait=1; // Ορισμός της μεταβλητής για την ενεργοποίηση του βρόχου ελέγχου
  int error;
  double dT;
  accel_t_gyro_union accel_t_gyro;

  // Κάλεισμα συνάρτησης για την συλλογή των ανεπεξέργαστων τιμών από την
  αδρανειακή μονάδα
  error = read_gyro_accel_vals((uint8_t*) &accel_t_gyro);

  // Χρόνος που έγινε η ανάγνωση για τον υπολογισμό των στροφών
  unsigned long t_now = millis();

  // Μετατροπή των τιμών του γυροσκοπίου σε degrees/sec
  float FS_SEL = 131;

  float gyro_x = (accel_t_gyro.value.x_gyro - base_x_gyro)/FS_SEL;
  float gyro_y = (accel_t_gyro.value.y_gyro - base_y_gyro)/FS_SEL;
  float gyro_z = (accel_t_gyro.value.z_gyro - base_z_gyro)/FS_SEL;

  // Ανάγνωση των ανεπεξέργαστων τιμών του επιταχυνσιόμετρου
  // float G_CONVERT = 16384;
  float accel_x = accel_t_gyro.value.x_accel;
  float accel_y = accel_t_gyro.value.y_accel;
  float accel_z = accel_t_gyro.value.z_accel;

  // Μετατροπή των τιμών σε γωνίες από το επιταχυνσιόμετρο

```

```

float RADIANS_TO_DEGREES = 180/3.14159;
float accel_vector_length = sqrt(pow(accel_x,2) + pow(accel_y,2) +
pow(accel_z,2));
float accel_angle_y = atan(-1*accel_x/sqrt(pow(accel_y,2) +
pow(accel_z,2)))*RADIANS_TO_DEGREES;
float accel_angle_x = atan(accel_y/sqrt(pow(accel_x,2) +
pow(accel_z,2)))*RADIANS_TO_DEGREES;

float accel_angle_z = 0;

// Υπολογισμών των φιλτραρισμένων γωνιών του γυροσκοπίου
float dt =(t_now - get_last_time())/1000.0;
float gyro_angle_x = gyro_x*dt + get_last_x_angle();
float gyro_angle_y = gyro_y*dt + get_last_y_angle();
float gyro_angle_z = gyro_z*dt + get_last_z_angle();

// Υπολογισμός των ολισθημένων γωνιών του γυροσκοπίου
float unfiltered_gyro_angle_x = gyro_x*dt + get_last_gyro_x_angle();
float unfiltered_gyro_angle_y = gyro_y*dt + get_last_gyro_y_angle();
float unfiltered_gyro_angle_z = gyro_z*dt + get_last_gyro_z_angle();

// Εφαρμογή του complementary φίλτρου για την εξακρίβωση αλλαγής στις
γωνίες. Επίσης γίνεται εκτίμηση του α το οποίο εξαρτάται από τον χρόνο
δειγματοληψίας.
float alpha = 0.96;
float angle_x = alpha*gyro_angle_x + (1.0 - alpha)*accel_angle_x;
float angle_y = alpha*gyro_angle_y + (1.0 - alpha)*accel_angle_y;
float angle_z = alpha*gyro_angle_z +(1.0 - alpha)*accel_angle_z;
// Το επιταχυνσιόμετρο δεν δίνει γωνία στον Z άξονα λόγω της βαρύτητας

// Ενημέρωση των αποθηκευμένων δεδομένων με τις τελευταίες μετρήσεις
set_last_read_angle_data(t_now, angle_x, angle_y, angle_z,
unfiltered_gyro_angle_x, unfiltered_gyro_angle_y, unfiltered_gyro_angle_z);

// Καθορισμός της δύναμης με την μετατροπή του χρόνου της τηλεκατεύθυνσης
σε επιθυμητή δύναμη ώσης
force=map(Rc_Throttle_ch,1072,1890,0,570);

// Καθορισμός της επιθυμητής γωνιακής θέσης στον Y άξονα με την μετατροπή
του χρόνου της τηλεκατεύθυνσης με όρια τις 10/-10 μοίρες
theta=map(Rc_Roll_ch,20780,19970,10,-10);

// Καθορισμός της επιθυμητής γωνιακής θέσης στον X άξονα με την μετατροπή
του χρόνου της τηλεκατεύθυνσης με όρια τις 10/-10 μοίρες
phi=map(Rc_Pitch_ch,20780,19970,10,-10);

// Έλεγχος για την διατήρηση του ορίου της επιθυμητής γωνιακής θέσης που
μπορεί να δώσει ο αρχάριος χρήστης με σκοπό την ομαλότερη πτήση. Για
έμπειρους χρήστες το όριο ανεβαίνει.
if(phi>10)
{

```

```

        phi=10;
    }
    if(phi<-10) // Έλεγχος για την διατήρηση του ορίου της επιθυμητής
γωνιακής θέσης.
    {
        phi=-10;
    }
    if(theta>10) // Έλεγχος για την διατήρηση του ορίου της επιθυμητής
γωνιακής θέσης.
    {
        theta=10;
    }
    if(theta<-10) // Έλεγχος για την διατήρηση του ορίου της επιθυμητής
γωνιακής θέσης.
    {
        theta=-10;
    }

// Επιθυμητές γωνιακές ταχύτητες για κάθε άξονα
desx=0;
desy=0;
desz=0;

////////////////////////////////////// ΕΛΕΓΚΤΗΣ Χ ΑΞΟΝΑ ////////////////////////////////////////

    phi_error=phi-angle_x; // Υπολογισμός σφάλματος γωνιακής θέσης στον X
άξονα
    rate_errorx=desx-(gyro_x); // Υπολογισμός σφάλματος γωνιακής ταχύτητας
στον X άξονα

    if(saturated_x==0) // Έλεγχος σημαίας κορεσμού του σήματος στον X
άξονα
    {
        if(phi_error>0) // Έλεγχος προσήμου του σφάλματος
        {
            Integral_x +=(abs(phi_error)*T); // Ολοκλήρωση του σφάλματος
στον X άξονα
        }
        if(phi_error<0) // Έλεγχος προσήμου του σφάλματος
        {
            Integral_x-=(abs(phi_error)*T); // Ολοκλήρωση του σφάλματος
στον X άξονα
        }
    }
    else
    {
        Integral_x=0; // Αν το σήμα είναι κορεσμένο το ολοκλήρωμα
μηδενίζεται
    }

```

```

// Υπολογισμός του σήματος από τον PI ελεγκτή
phi_signal=phi_error*Kpx+Integral_x*Kix;

// Υπολογισμός του σήματος από τον D ελεγκτή
rate_signalx=rate_errorx*Kdx;

// Υπολογισμός της επιθυμητής δύναμης από τον PI-D ελεγκτή
rate_x=phi_signal+(rate_signalx);

// Έλεγχος για κορεσμό της επιθυμητής δύναμης στον X άξονα στο κάτω όριο
if(rate_x<-150)
{
    saturated_x=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_x=-150; // Ορισμός της επιθυμητής δύναμης στο κάτω όριο
}

// Έλεγχος για κορεσμό της επιθυμητής δύναμης στον X άξονα στο άνω όριο
else if(rate_x>150)
{
    saturated_x=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_x=150; // Ορισμός της επιθυμητής δύναμης στο άνω όριο
}
else
{
    // Ορισμός της σημαίας στο 0 γιατί το σήμα δεν είναι κορεσμένο
    saturated_x=0;
}

////////////////////////////////////// ΕΛΕΓΚΤΗΣ Υ ΑΞΟΝΑ ////////////////////////////////////////

// Υπολογισμός σφάλματος γωνιακής θέσης στον Y άξονα
theta_error=theta-angle_y;

// Υπολογισμός σφάλματος γωνιακής ταχύτητας στον Y άξονα
rate_error_y=desy-(gyro_y);

// Έλεγχος σημαίας κορεσμού του σήματος στον Y άξονα
if(saturated_y==0)
{
    if(theta_error>0) // Έλεγχος προσήμου του σφάλματος
    { // Ολοκλήρωση του σφάλματος στον Y άξονα
        Integral_y +=(abs(theta_error)*T);
    }
    if(theta_error<0) // Έλεγχος προσήμου του σφάλματος
    { // Ολοκλήρωση του σφάλματος στον Y άξονα
        Integral_y -= (abs(theta_error)*T);
    }
}
else // Αν το σήμα είναι κορεσμένο το ολοκλήρωμα μηδενίζεται
{

```

```

    Integral_y=0;
}

// Υπολογισμός του σήματος από τον PI ελεγκτή
theta_signal=theta_error*Kpy+Integral_y*Kiy;

// Υπολογισμός του σήματος από τον D ελεγκτή
rate_signaly=rate_errory*Kdy;

// Υπολογισμός της επιθυμητής δύναμης από τον PI-D ελεγκτή
rate_y=theta_signal+(rate_signaly);

// Έλεγχος για κορεσμό της επιθυμητής δύναμης στον Y άξονα στο κάτω όριο
if(rate_y<-150)
{
    saturated_y=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_y=-150; // Ορισμός της επιθυμητής δύναμης στο κάτω όριο
}

// Έλεγχος για κορεσμό της επιθυμητής δύναμης στον Y άξονα στο άνω όριο
else if(rate_y>150)
{
    saturated_y=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_y=150; // Ορισμός της επιθυμητής δύναμης στο άνω όριο
}
else
{
    saturated_y=0; // Ορισμός της σημαίας στο 0 γιατί το σήμα δεν είναι
κορεσμένο
}

////////////////////////////////// ΕΛΕΓΚΤΗΣ Z ΑΞΟΝΑ ////////////////////////////////////

if(saturated_z==0) // Έλεγχος σημαίας κορεσμού του σήματος στον Z άξονα
{
    if(psi_error>0) // Έλεγχος για το πρόσημο του σφάλματος στον Z άξονα
    {
        Integral_z +=(abs(psi_error)*T); // Ολοκλήρωση του σφάλματος
    }
    if(psi_error<0)
    {
        Integral_z -= (abs(psi_error)*T); // Ολοκλήρωση του σφάλματος
    }
}
else
{
    Integral_z=0; // Αν το σήμα είναι κορεσμένο τότε το ολοκλήρωμα
μηδενίζεται
}

// Υπολογισμός σφάλματος της γωνιακής θέσης στον Z άξονα

```

```

psi_error=psi-gyro_z;

// Υπολογισμός του σφάλματος της γωνιακής ταχύτητας στον Z άξονα
rate_errorz=desz-(gyro_z);

// Υπολογισμός του σήματος από τον PI ελεγκτή
psi_signal=psi_error*Kpz+Integral_z*Kiz;

// Υπολογισμός της δύναμης για τον έλεγχο της γωνιακής ταχύτητας στον Z
// άξονα
rate_z=psi_signal + (rate_errorz*Kdz);

// Έλεγχος για κορεσμό του σήματος στον Z άξονα στο άνω όριο
if(rate_z>50)
{
    saturated_z=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_z=50; // Όρισμός της επιθυμητής δύναμης στο όριο
}
// Έλεγχος για κορεσμό του σήματος στον Z άξονα στο κάτω όριο
else if(rate_z<-50)
{
    saturated_z=1; // Ενεργοποίηση σημαίας κορεσμού του σήματος
    rate_z=-50; // Όρισμός της επιθυμητής δύναμης στο όριο
}
else
{
    saturated_z=0; // Όρισμός της σημαίας στο 0 γιατί το σήμα δεν είναι
    κορεσμένο
}

////////////////////////////////////

float force_rate_y=force+rate_y+rate_z; // Συνδιασμός των επιθυμητών
δύναμεων ώσης/περιστροφής για τον κινητήρα 4 από τους ελεγκτές

float force_rate_y1=force-rate_y+rate_z; // Συνδιασμός των επιθυμητών
δύναμεων ώσης/περιστροφής για τον κινητήρα 2 από τους ελεγκτές

float force_rate_x=force-rate_x+(-rate_z); // Συνδιασμός των επιθυμητών
δύναμεων ώσης/περιστροφής για τον κινητήρα 3 από τους ελεγκτές

float force_rate_x1=force+rate_x+(-rate_z); // Συνδιασμός των επιθυμητών
δύναμεων ώσης/περιστροφής για τον κινητήρα 1 από τους ελεγκτές

//////////////////////////////////// X AXIS MOTOR SIGNALS //////////////////////////////////////

// Υπολογισμός της δύναμης με το πολυόνομο για τον κινητήρα 3 και
καθορισμός του OCR4A για την παραγωγή του ανάλογου PWM
OCR1B=20900-
(((m3_x3*force_rate_x1*force_rate_x1*force_rate_x1)+(m3_x2*force_rate_x1*fo
rce_rate_x1)+(m3_x1*force_rate_x1)-m3_x0)*9.4);

```

```

// Υπολογισμός της δύναμης με το πολυώνυμο για τον κινητήρα 8 και
καθορισμός του OCR4A για την παραγωγή του ανάλογου PWM
OCR4A=20900-
((m8_x3*force_rate_x*force_rate_x*force_rate_x)+(m8_x2*force_rate_x*force_
rate_x)+(m8_x1*force_rate_x)-m8_x0)*9.4);

////////////////////////////////////// Y AXIS MOTOR SIGNALS ////////////////////////////////////////

// Υπολογισμός της δύναμης με το πολυώνυμο για τον κινητήρα 2 και
καθορισμός του OCR1C για την παραγωγή του ανάλογου PWM
OCR1C=20900-
((m6_x3*force_rate_y1*force_rate_y1*force_rate_y1)+(m6_x2*force_rate_y1*fo
rce_rate_y1)+(m6_x1*force_rate_y1)-m6_x0)*9.4);

// Υπολογισμός της δύναμης με το πολυώνυμο για τον κινητήρα 4 και
καθορισμός του OCR1A για την παραγωγή του ανάλογου PWM
OCR1A=20900-
((m7_x3*force_rate_y*force_rate_y*force_rate_y)+(m7_x2*force_rate_y*force_
rate_y)+(m7_x1*force_rate_y)-m7_x0)*9.4);

while(wait==1); // Διατήρηση του βρόχου μέχρι η μεταβλητή να γίνει μηδέν
}

//////////////////////////////////////

ISR(TIMER5_COMPA_vect) // Συνάρτηση ταύτισης συγκριτή του χρονιστή 5
{
    wait=0; // Έτσι εξασφαλίζουμε τον σταθερό χρόνο του βρόχου ελέγχου
}

// Συνάρτηση για την μέτρηση του χρόνου του σήματος για το κανάλι Pitch της
τηλεκατεύθυνσης
void Pitch_isr()
{
    Pitch_counter++; // Αύξηση του μετρητή

    // Έλεγχος για την πρώτη φορά που ενεργοποιείται η συνάρτηση
    if (Pitch_counter==1 )
    {
        Rc_Pitch_flag1= micros(); // Χρόνος που το σήμα κατεβαίνει στα 0V
    }
    // Έλεγχος για την δεύτερη φορά που ενεργοποιείται η συνάρτηση
    if(Pitch_counter==2)
    {
        Rc_Pitch_flag2 = micros();// Χρόνος που το σήμα ανεβαίνει στα 5V
    }
}

```

```

        Rc_Pitch_ch = Rc_Pitch_flag2 - Rc_Pitch_flag1; // Υπολογισμός του
χρόνου του σήματος
        Pitch_counter=0; // Μηδενισμός του μετρητή
    }
}

```

// Συνάρτηση για την μέτρηση του χρόνου του σήματος για το κανάλι Throttle της τηλεκατεύθυνσης

```

void Throttle_isr()
{
    Throttle_counter++;
    if ( Throttle_counter==1 )
    {
        Rc_Thr_flag1= micros();
    }
    if(Throttle_counter==2)
    {
        Rc_Thr_flag2 = micros();
        Rc_Throttle_ch = Rc_Thr_flag2 - Rc_Thr_flag1;
        Throttle_counter=0;
    }
}

```

// Συνάρτηση για την μέτρηση του χρόνου του σήματος για το κανάλι Roll της τηλεκατεύθυνσης

```

void Roll_isr()
{
    Roll_counter++;
    if ( Roll_counter==1 )
    {
        Rc_Roll_flag1= micros();
    }
    if(Roll_counter==2)
    {
        Rc_Roll_flag2 = micros();
        Rc_Roll_ch = Rc_Roll_flag2 - Rc_Roll_flag1;
        Roll_counter=0;
    }
}

```

////////////////////////////////// Συναρτήσεις της Αδρανειακής Μονάδας //////////////////////////////////

```

int MPU6050_read(int start, uint8_t *buffer, int size)
{
    int i, n, error;

    Wire.beginTransaction(MPU6050_I2C_ADDRESS);
    n = Wire.write(start);
    if (n != 1)
        return (-10);
}

```



```

n = Wire.endTransmission(false); // Κράτα το I2C-bus
if (n != 0)
    return (n);

// Τριτη παράμετρος είναι αληθής απελευθέρωσε το I2C-bus αφού έχει γίνει
η ανάγνωση των δεδομένων.
Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);
i = 0;
while(Wire.available() && i<size)
{
    buffer[i++]=Wire.read();
}
if ( i != size)
    return (-11);

return (0); // return : no error
}

int MPU6050_write(int start, const uint8_t *pData, int size)
{
    int n, error;

    Wire.beginTransaction(MPU6050_I2C_ADDRESS);
    n = Wire.write(start); // Εγγραφή διεύθυνσης
    if (n != 1)
        return (-20);

    n = Wire.write(pData, size); // Εγγραφή bytes δεδομένων
    if (n != size)
        return (-21);

    error = Wire.endTransmission(true); // Απελευθέρωση I2C-bus
    if (error != 0)
        return (error);

    return (0); // Επιστροφή χωρίς σφάλμα
}

int MPU6050_write_reg(int reg, uint8_t data)
{
    int error;

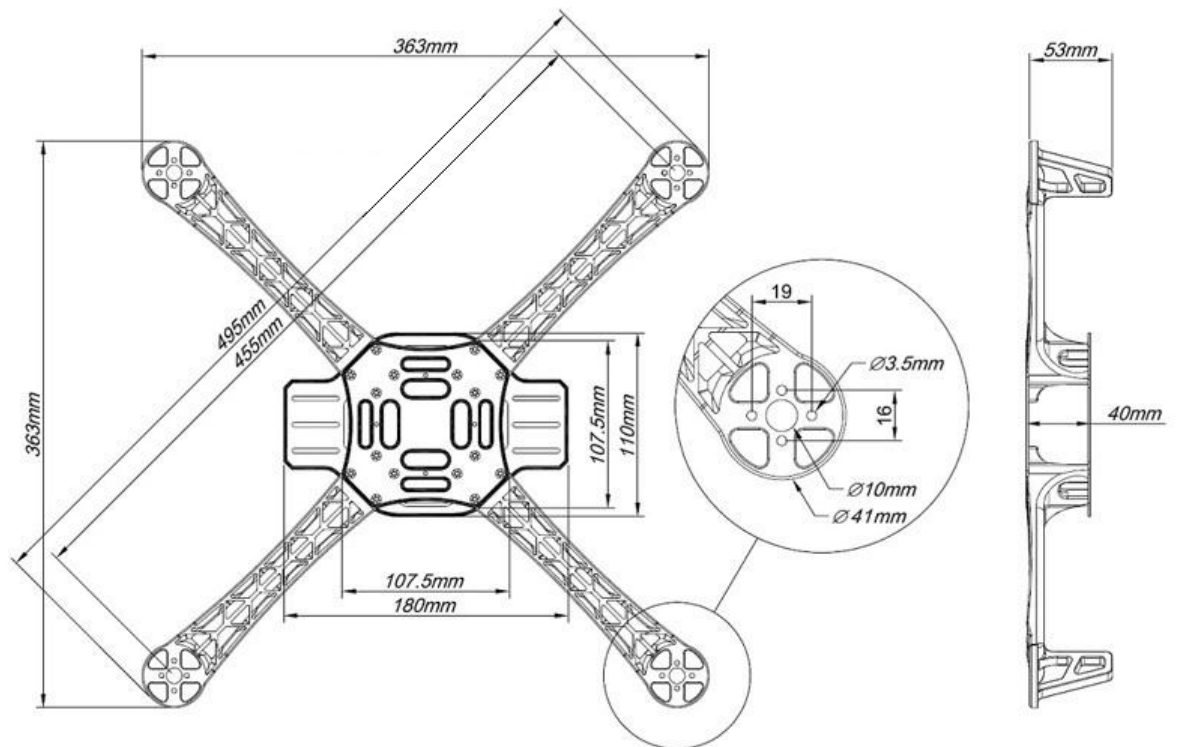
    error = MPU6050_write(reg, &data, 1);

    return (error);}

```


Παράρτημα-B: Σχηματικά Διαγράμματα

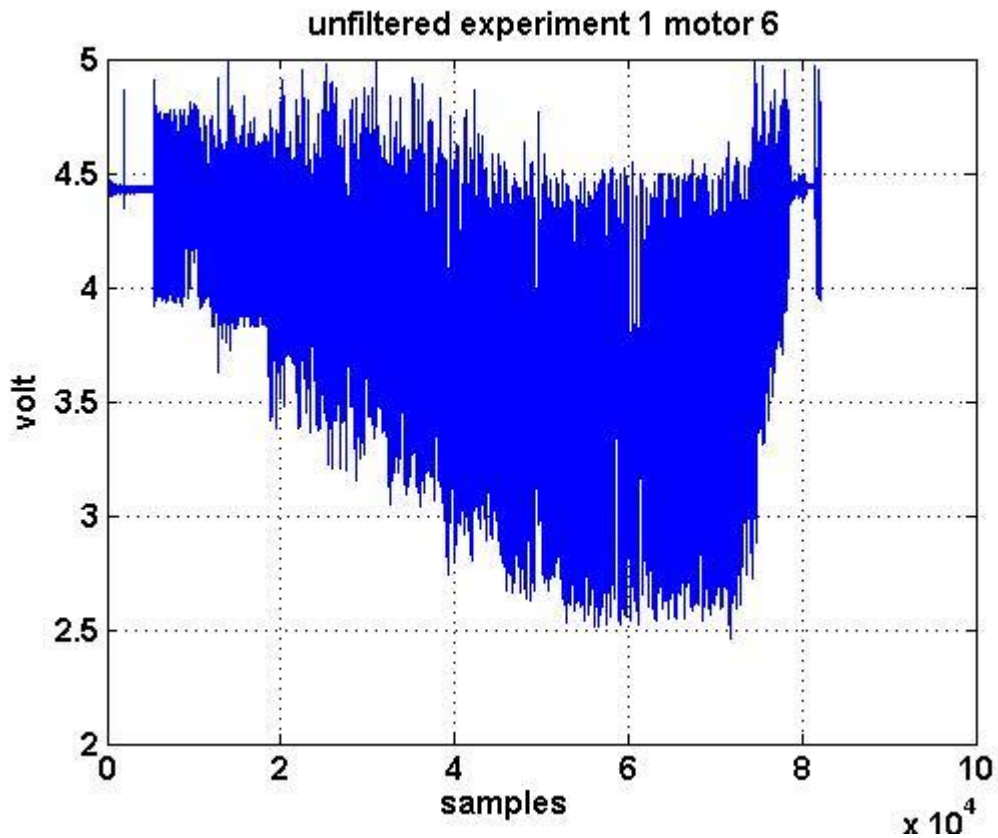
F450 πλαίσιο υπάμενης πλατφόρμας

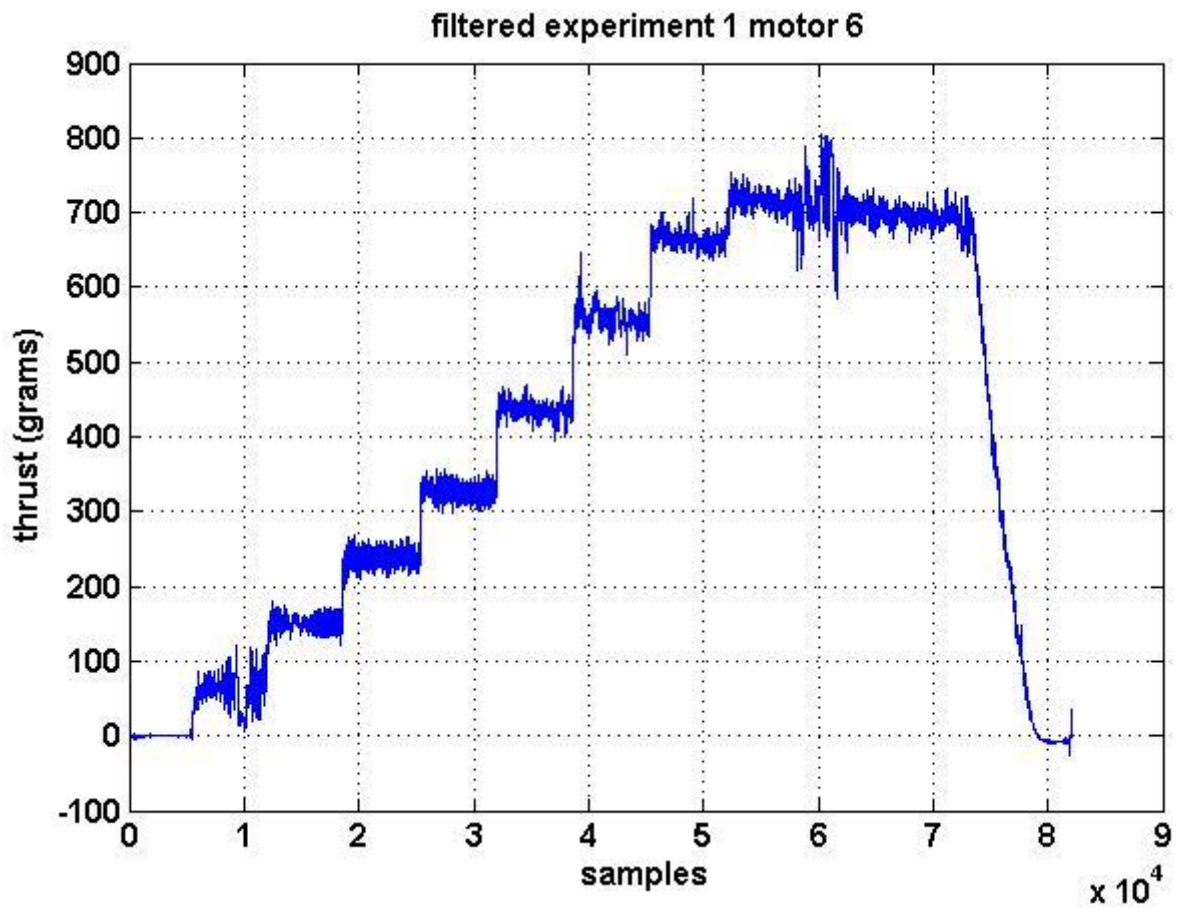


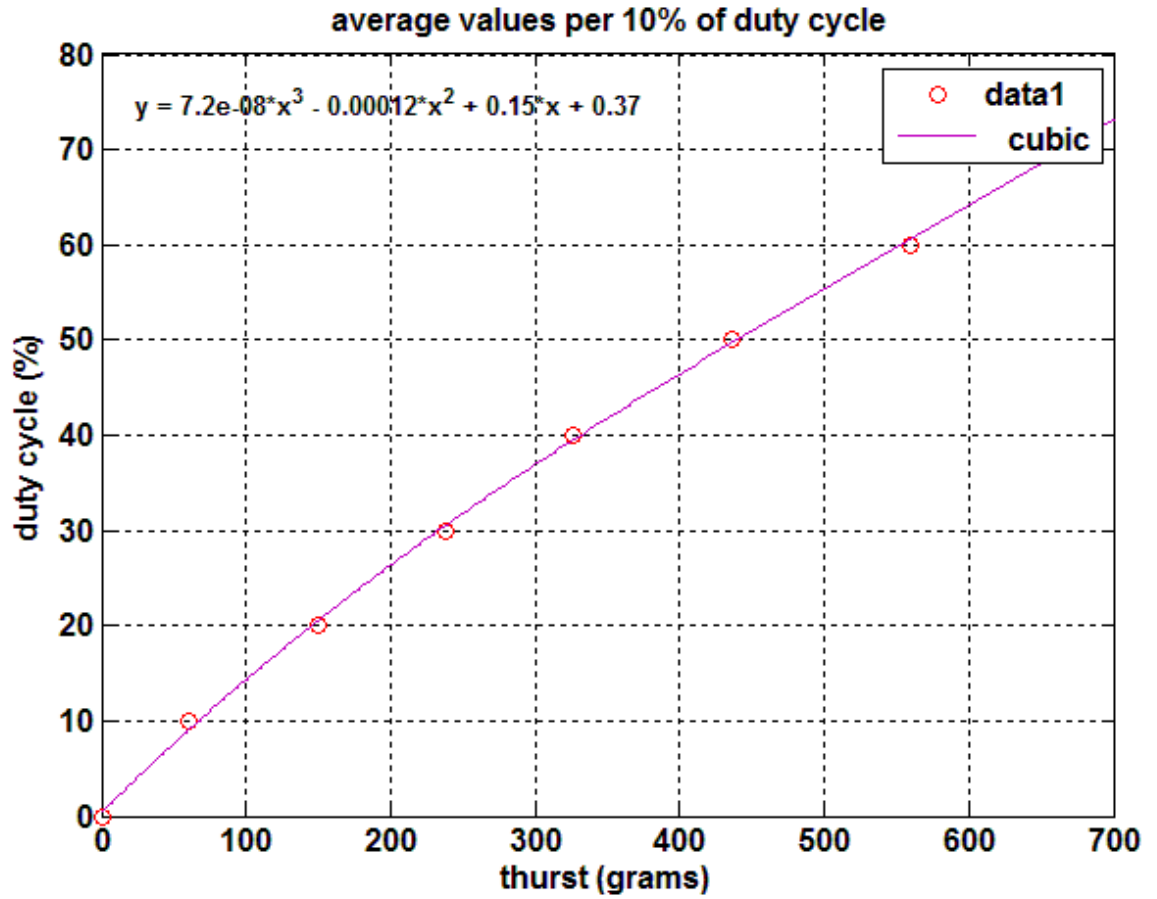
Παράρτημα-Γ: Πειράματα βαθμονόμησης

Σε αυτό το παράρτημα παρουσιάζονται τα γραφήματα από τα πειράματα που έγιναν για κάθε κινητήρα :

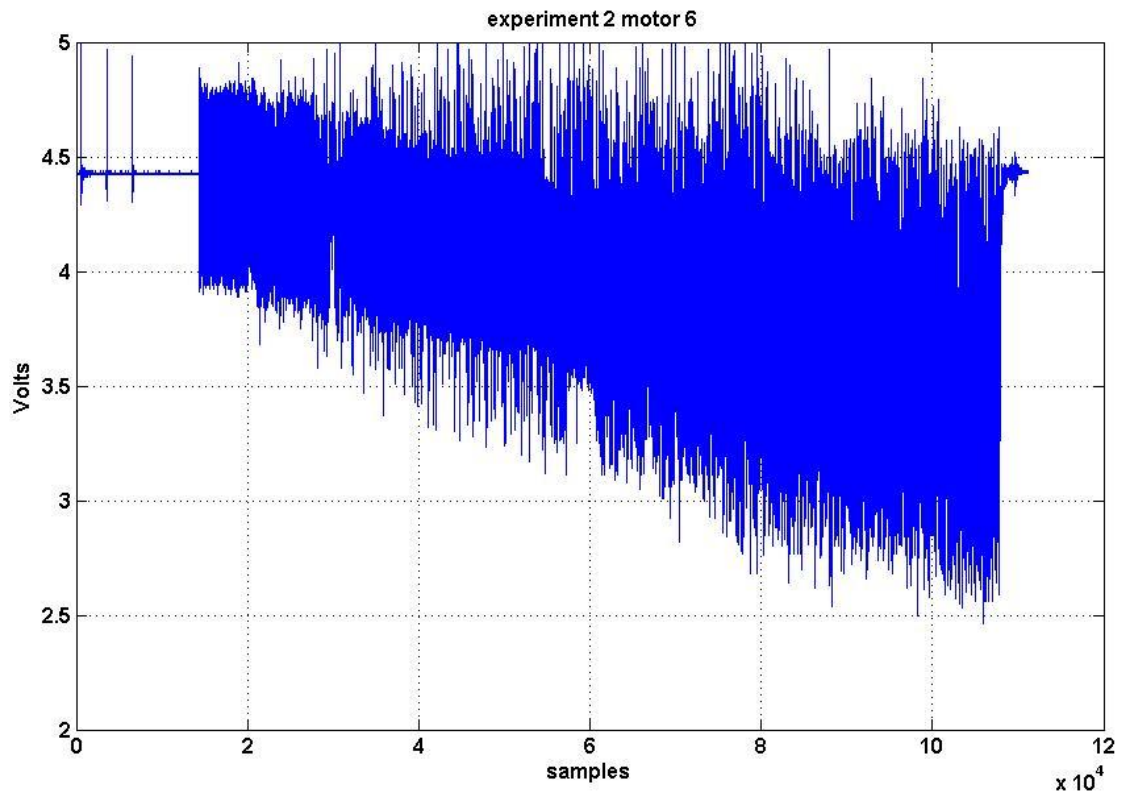
Πείραμα 1 Κινητήρας 6

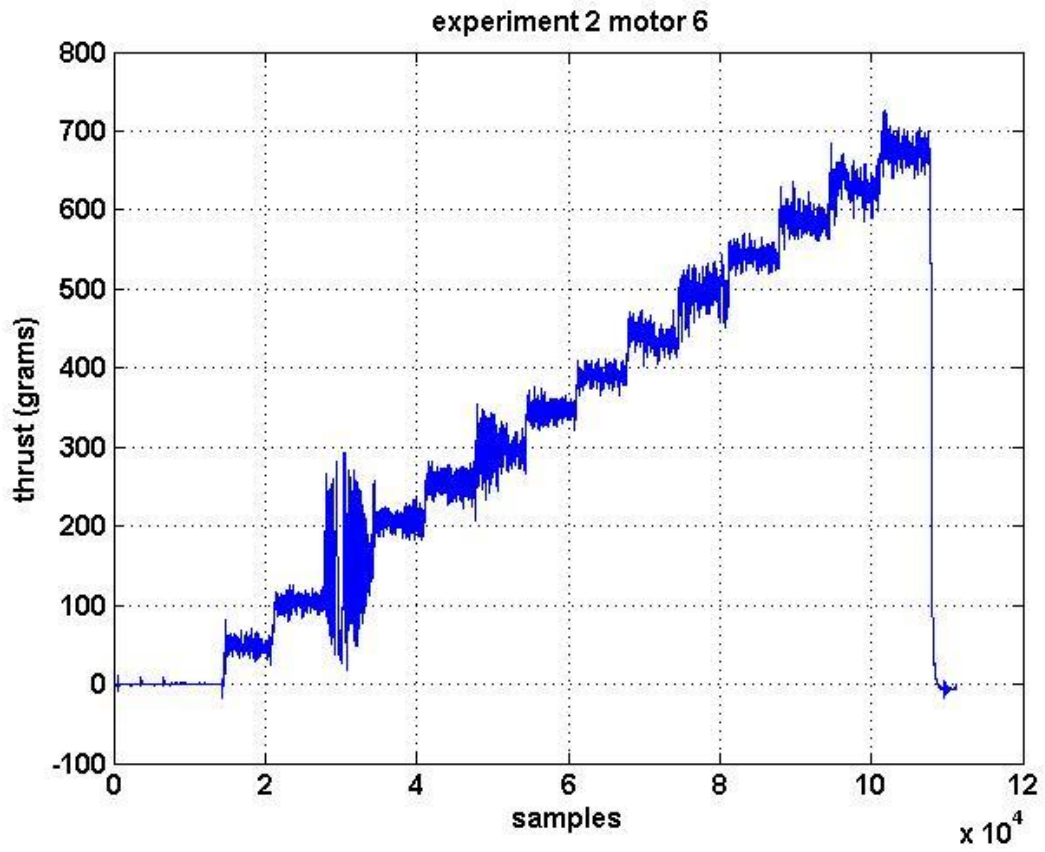






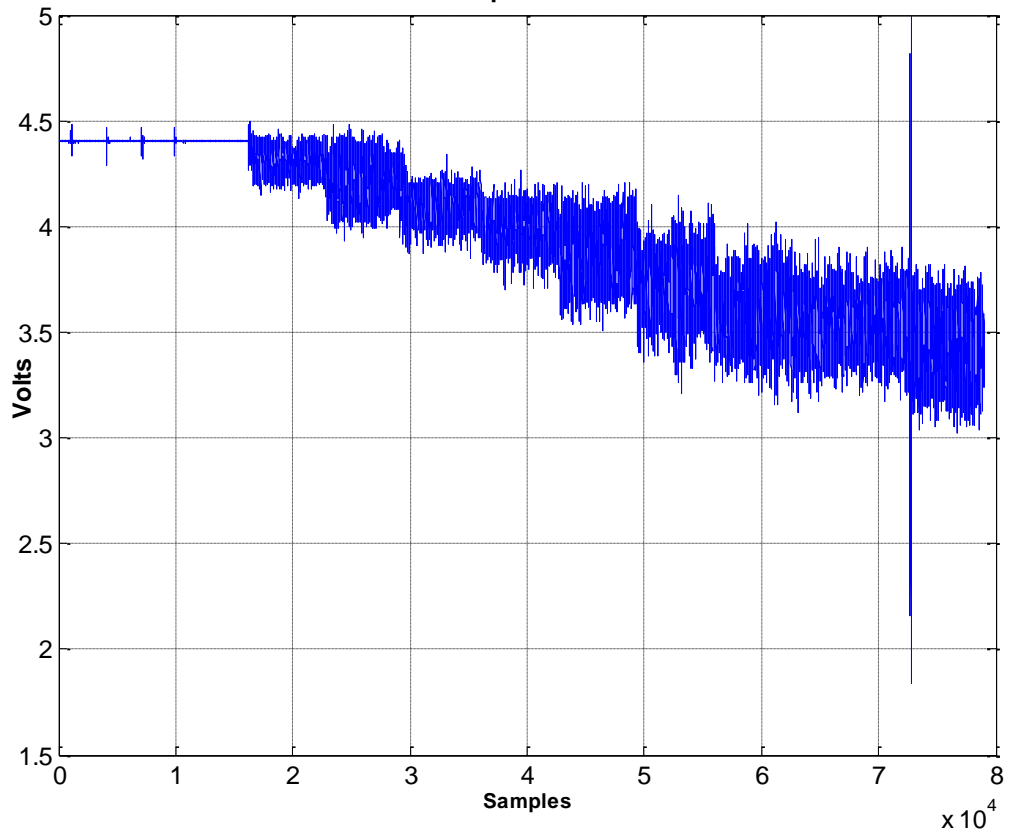
Πείραμα 2 κινητήρας 6



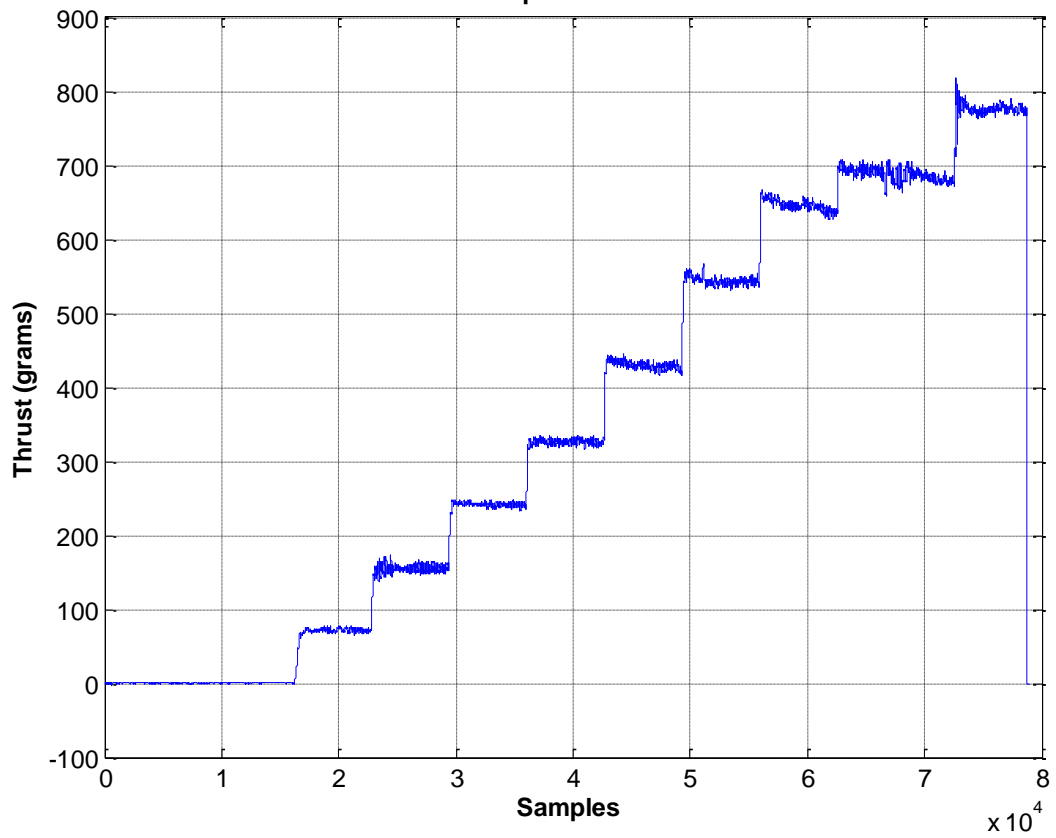


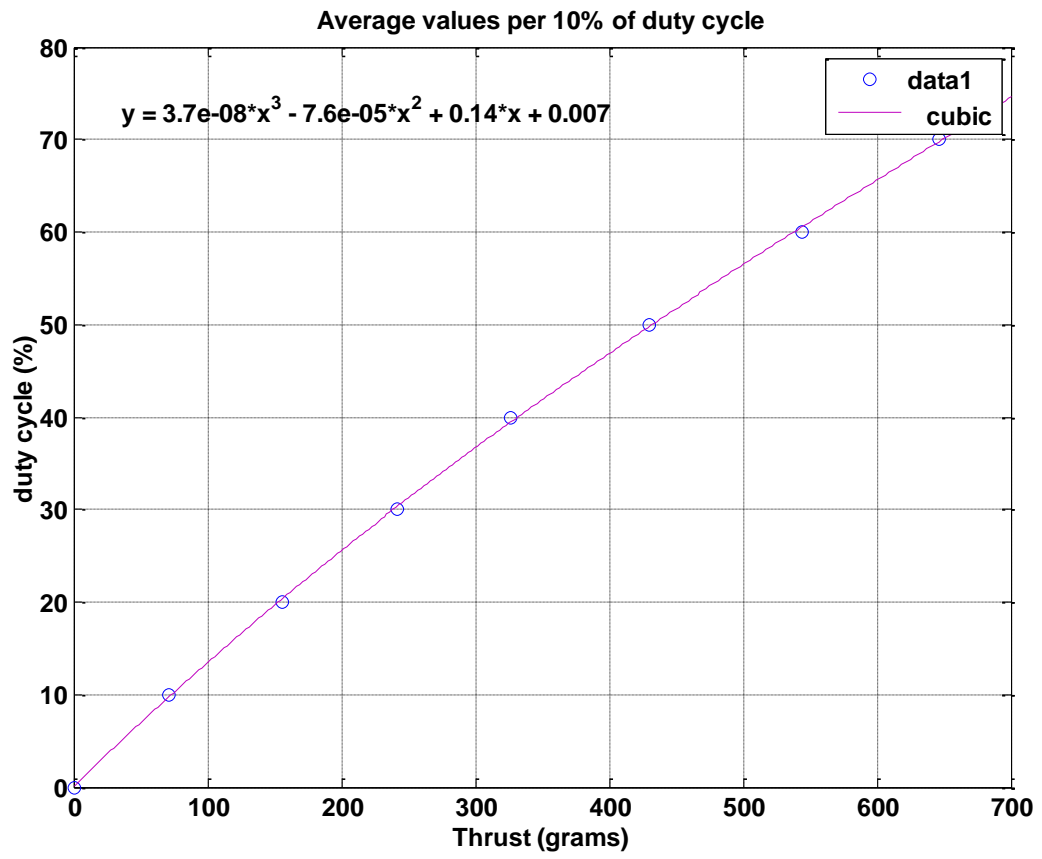
Πείραμα 1 Κινητήρας 7

Unfiltered expirement 1 motor 7



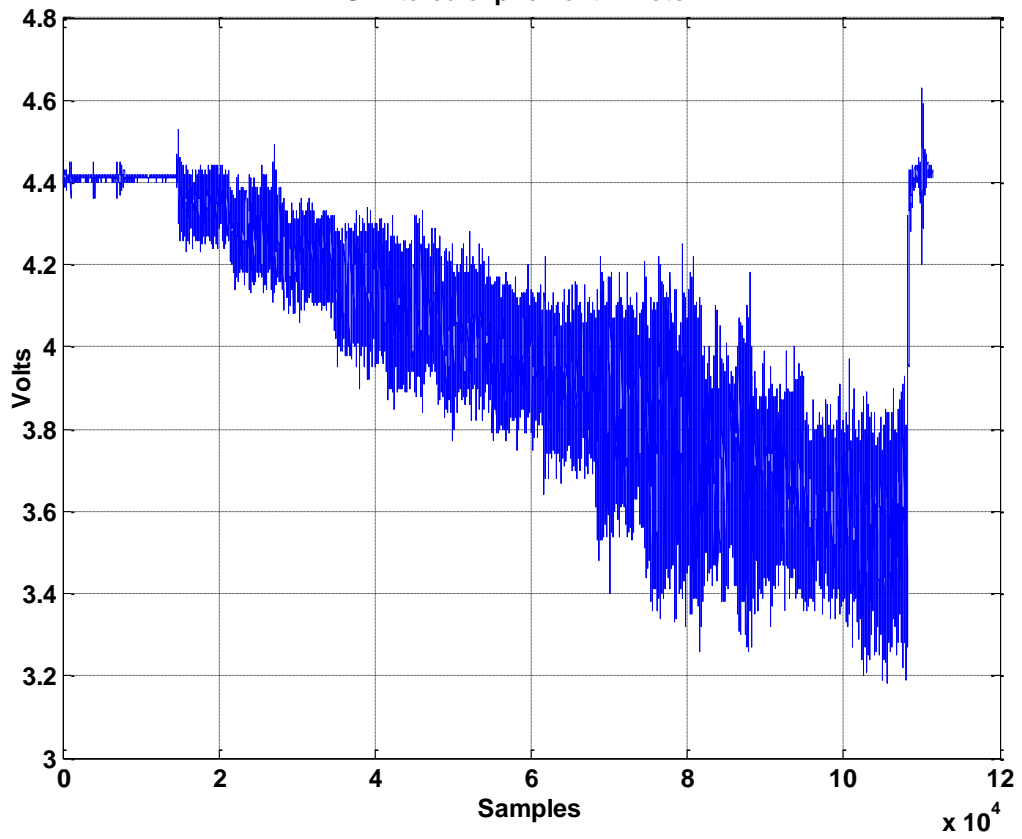
Filtered expirement 1 motor 7



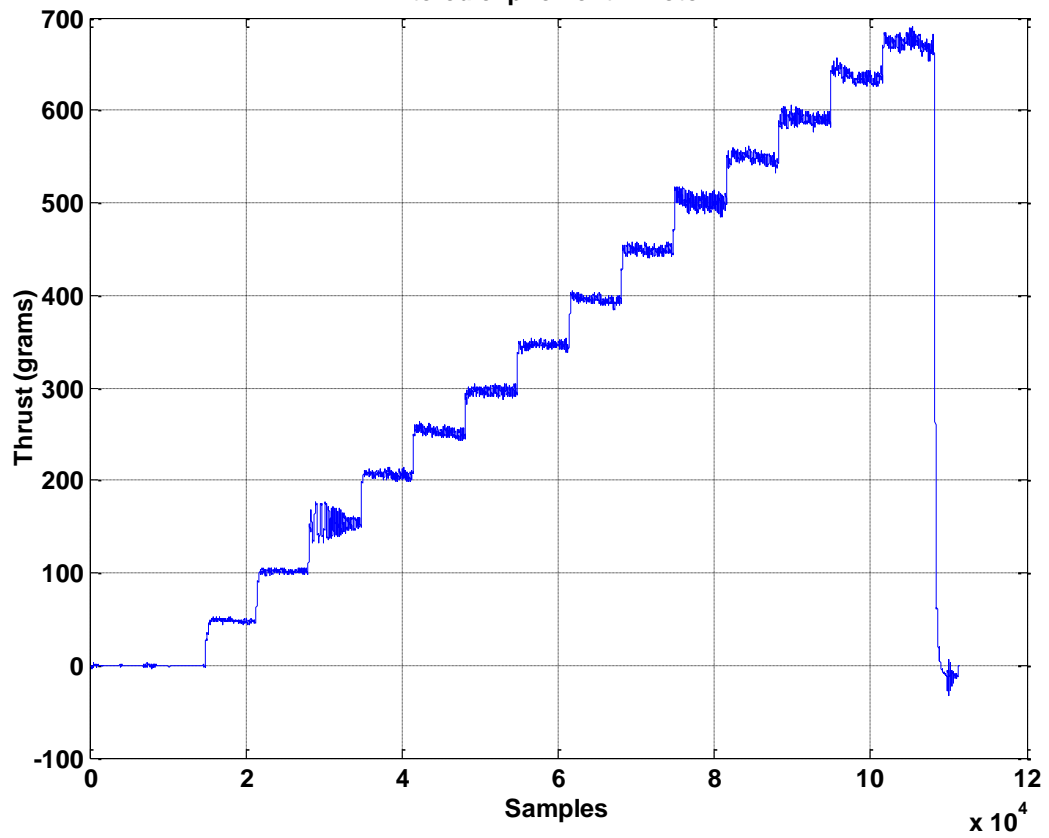


Πείραμα 2 Κινητήρας 7

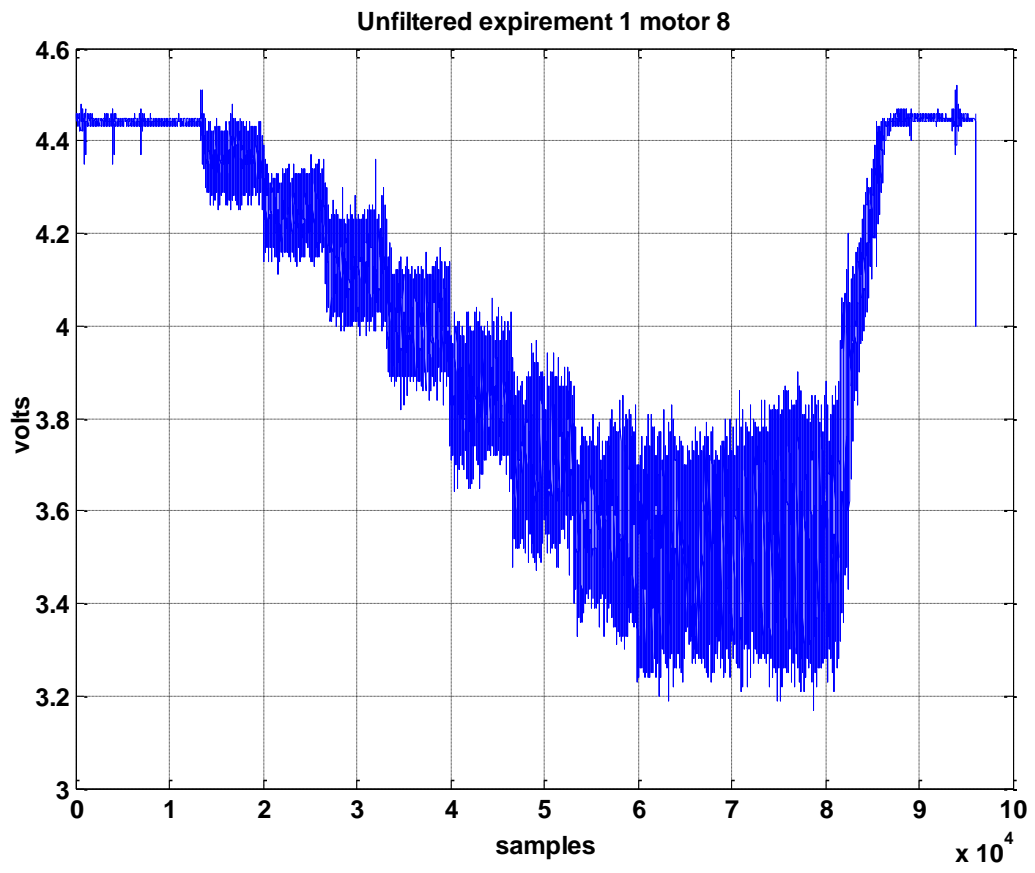
Unfiltered expirement 2 motor 7

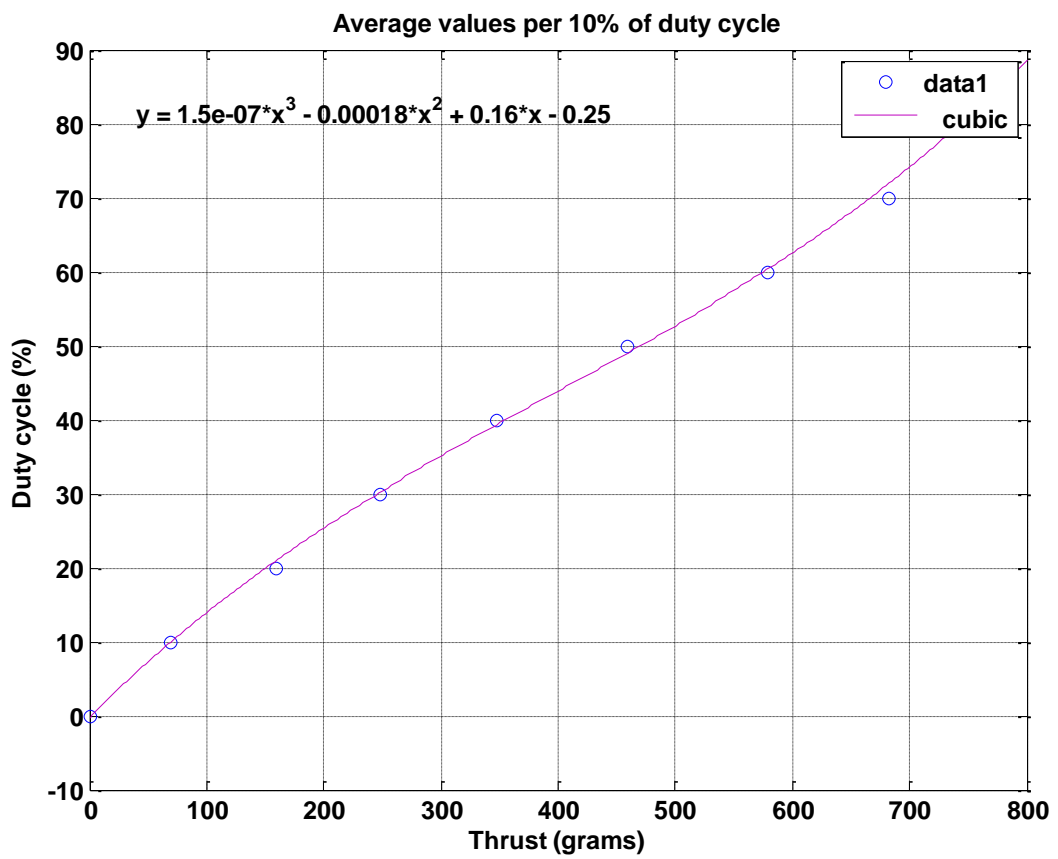
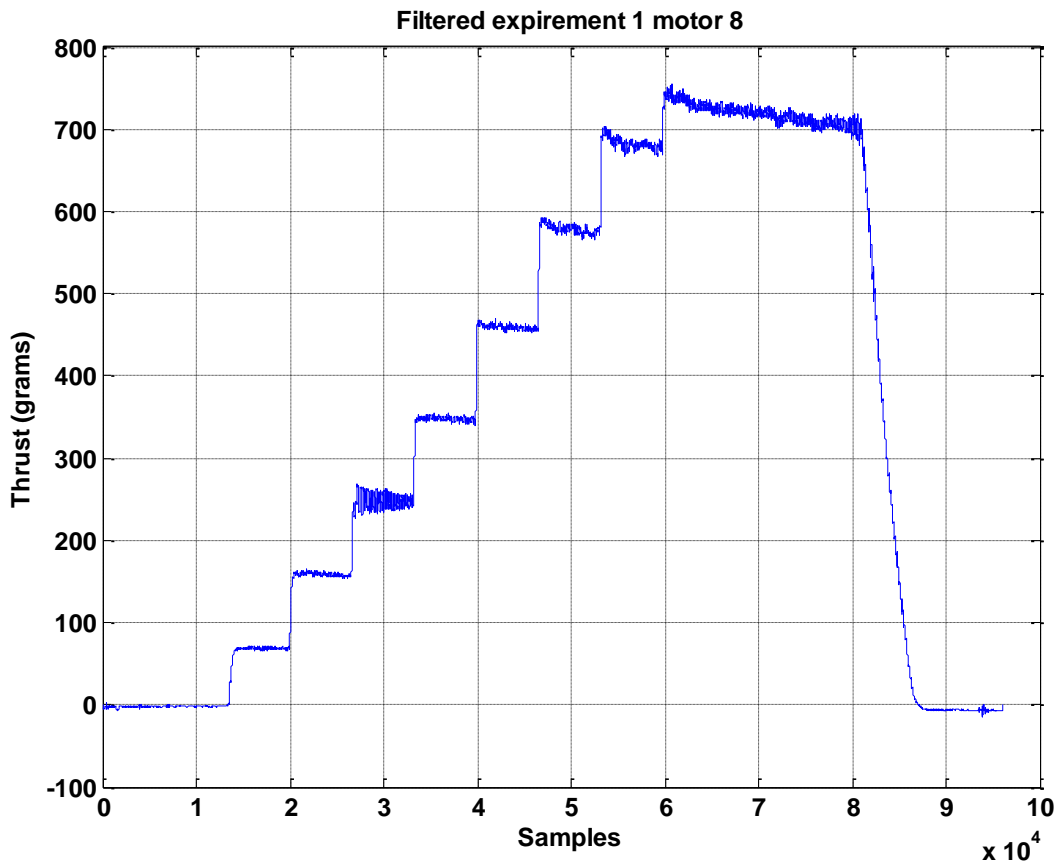


Filtered expirement 2 motor 7

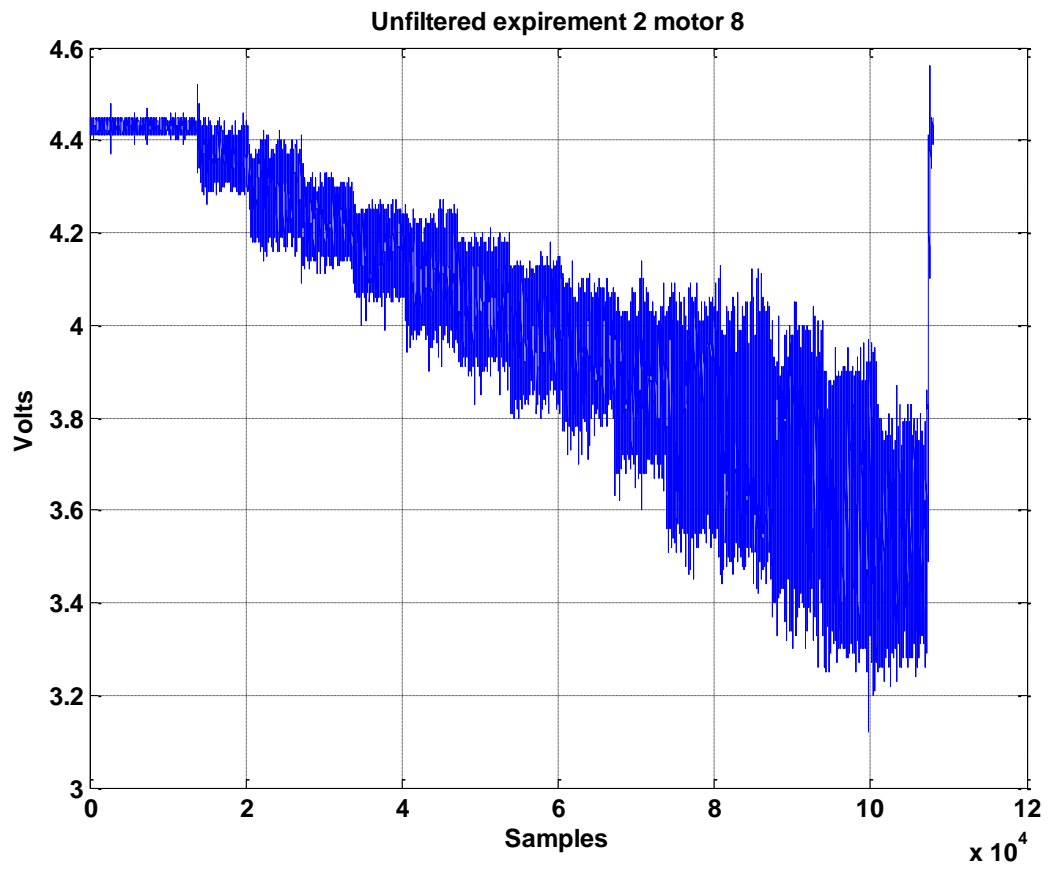


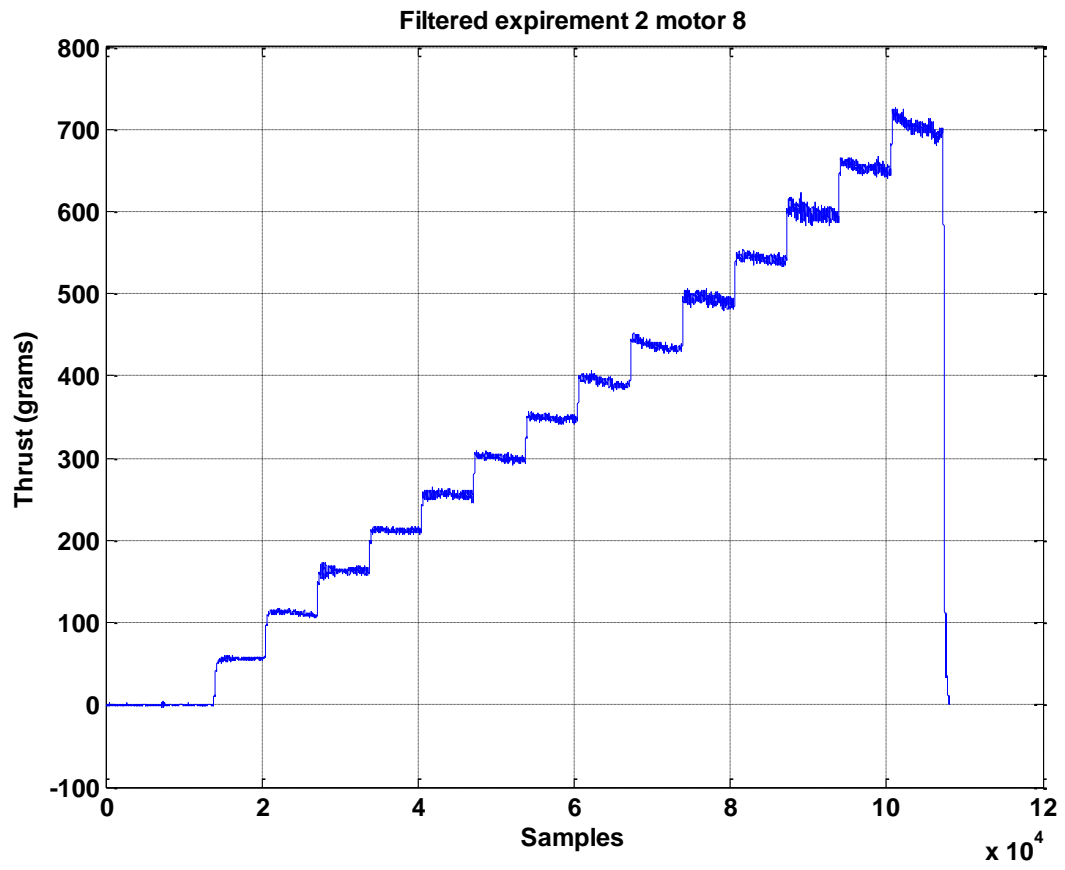
Πείραμα 1 Κινητήρας 8





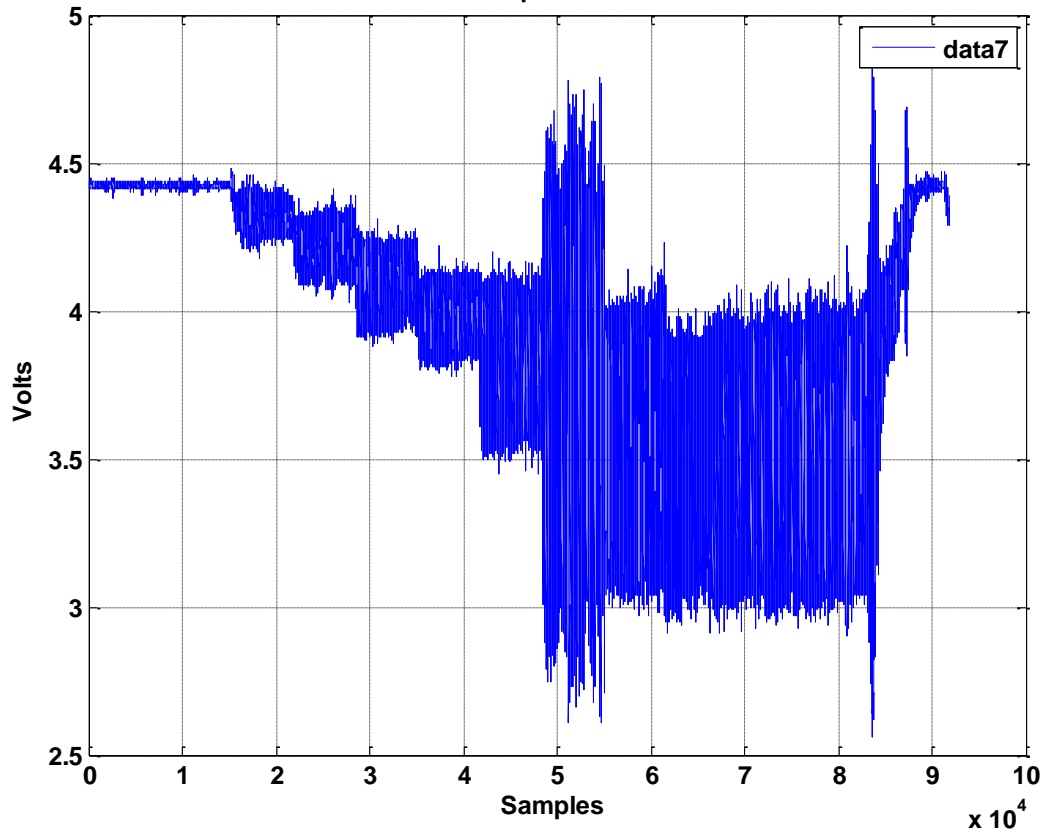
Πείραμα 2 Κινητήρας 8



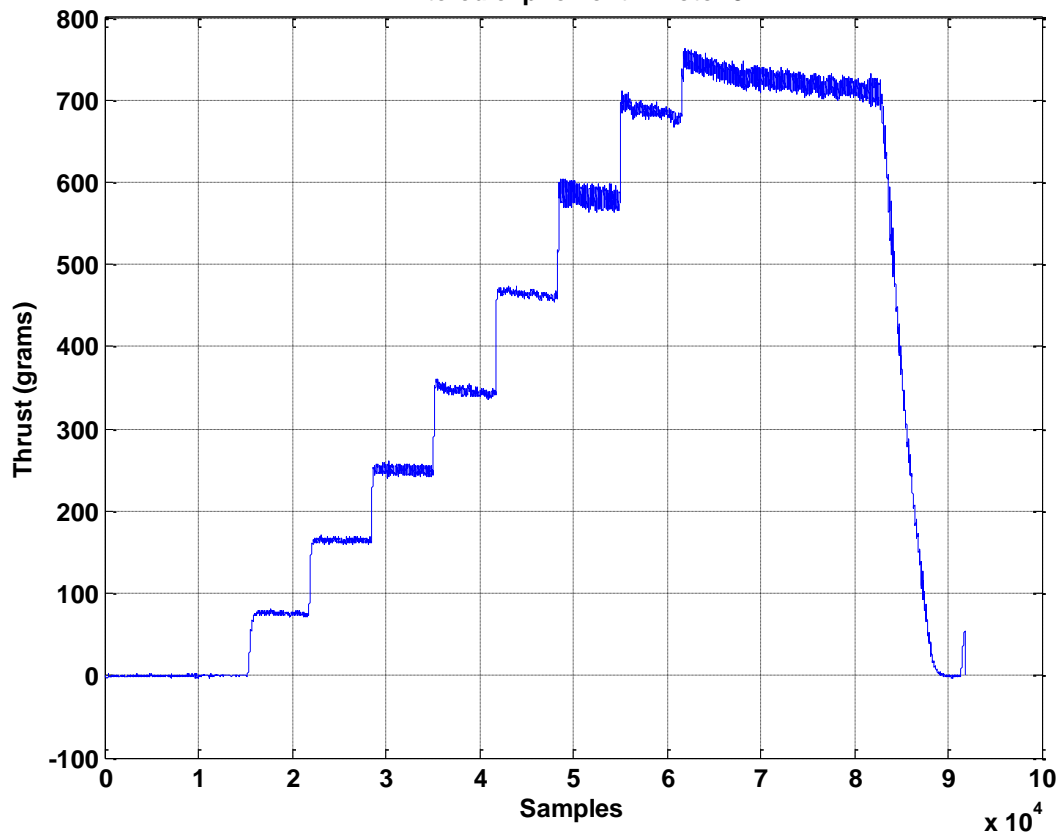


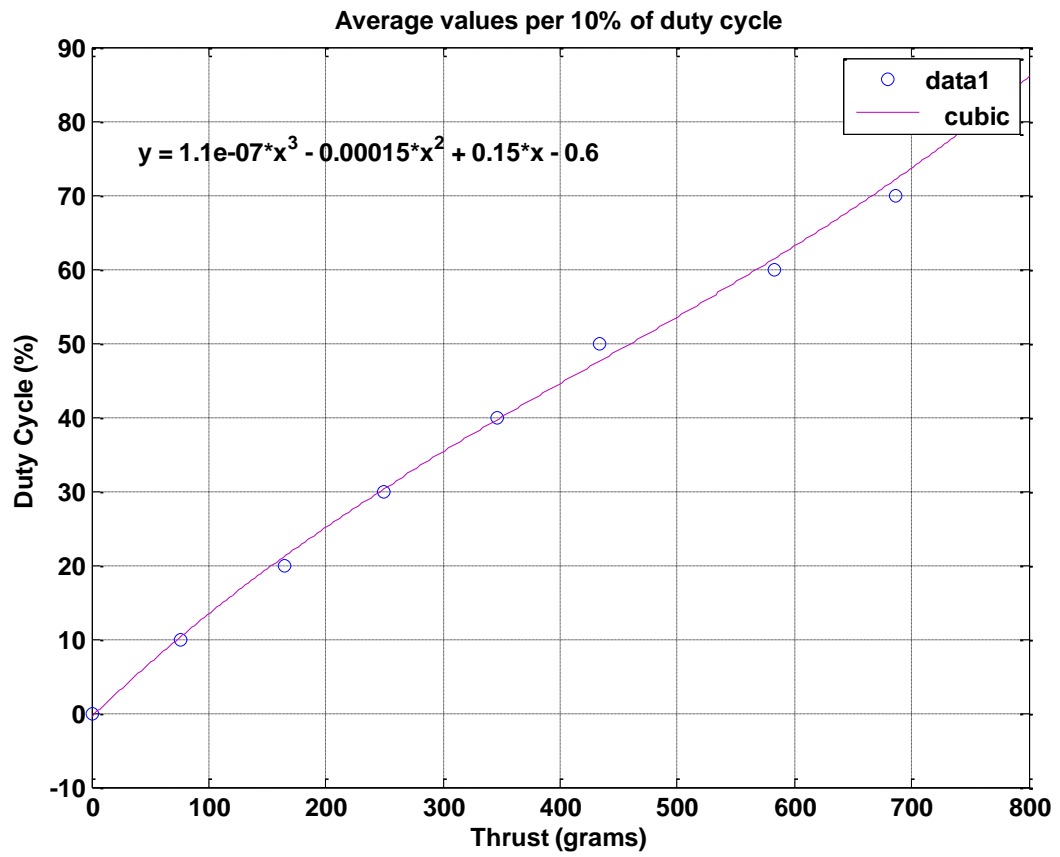
Πείραμα 1 Κινητήρας 9

Unfiltered expirement 1 motor 9



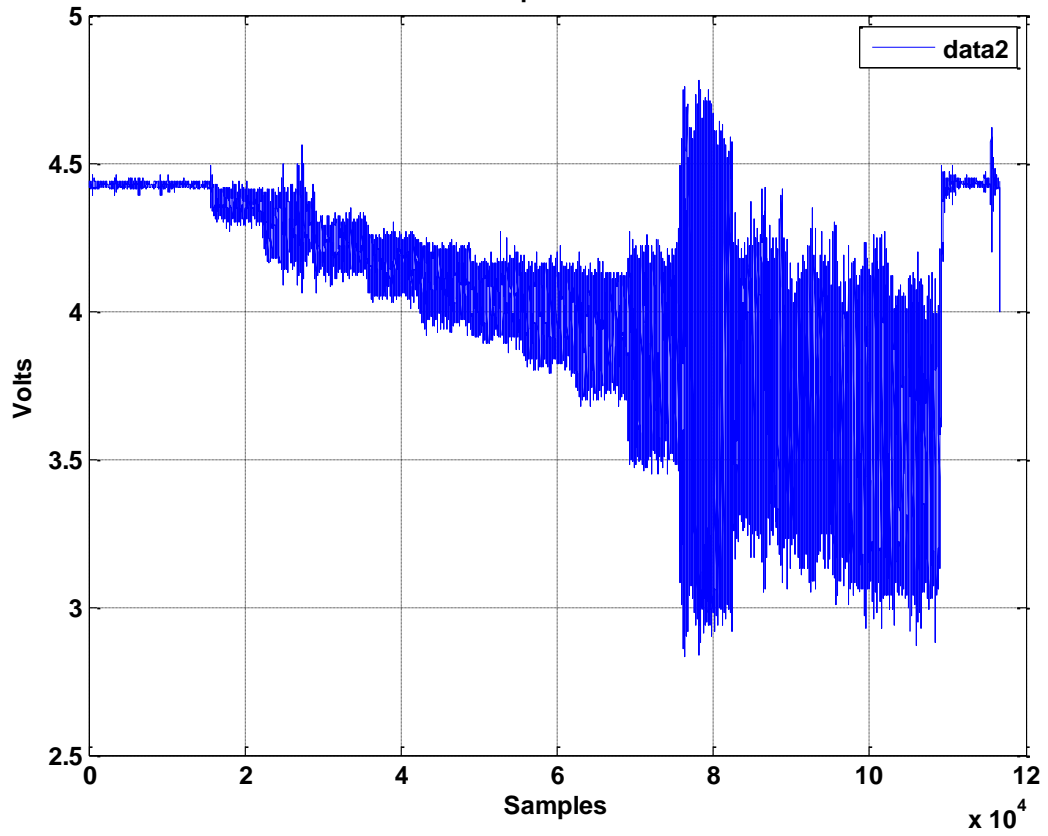
Filtered expirement 1 motor 9



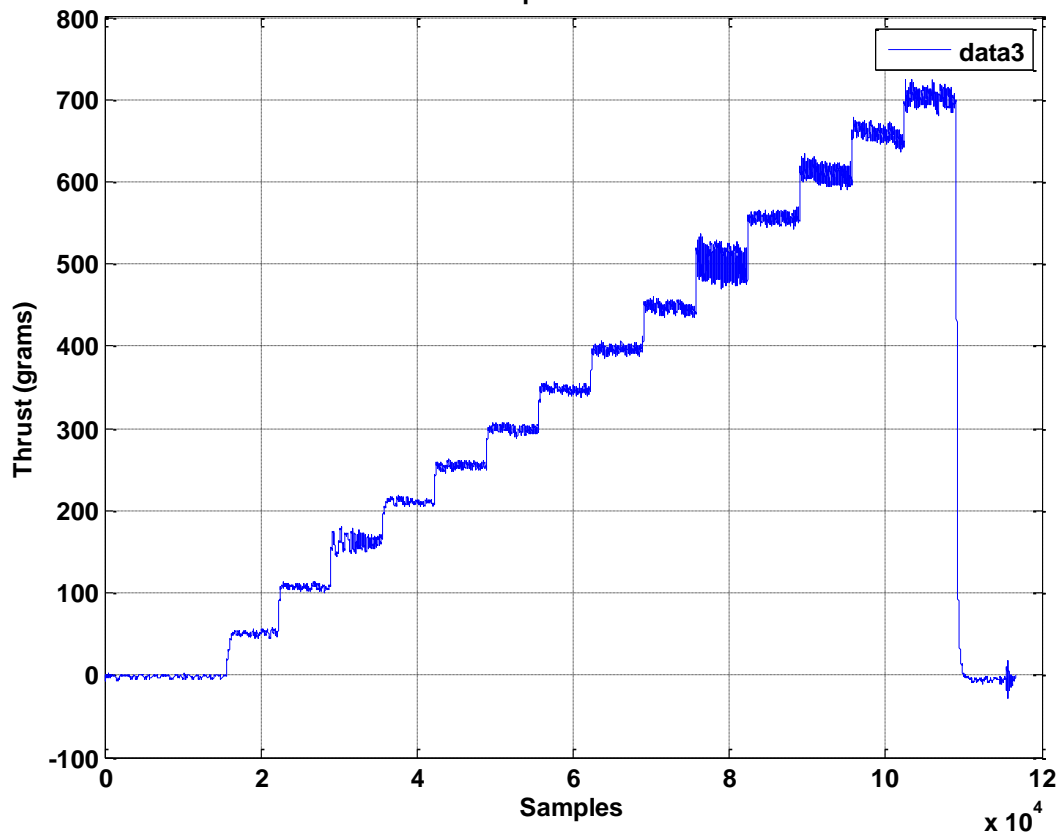


Πείραμα 2 Κινητήρας 9

Unfiltered expirement 2 motor 9



Filtered expirement 2 motor 9

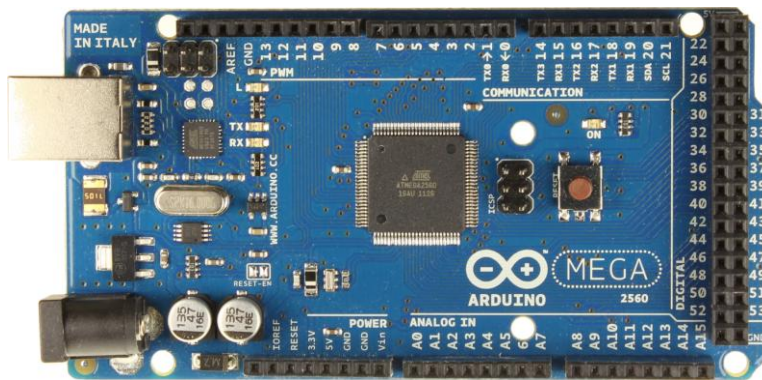


Παράρτημα-Δ:

Datasheets

Σε αυτό το παράρτημα παρατίθενται τα τεχνικά χαρακτηριστικά για συγκεκριμένα επιμέρους στοιχεία που χρησιμοποιούνται στην πειραματική διάταξη.

Arduino Mega 2560



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Mega 2560 is an update to the [Arduino Mega](#), which it replaces.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 boards) programmed as a USB-to-serial converter. Revision 2 of the Mega2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

Revision 3 of the board has the following new features:

1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

Stronger RESET circuit.

Atmega 16U2 replace the 8U2.

Schematic, Reference Design & Pin Mapping

EAGLE files: [arduino-mega2560_R3-reference-design.zip](#)

Schematic: [arduino-mega2560_R3-schematic.pdf](#)

Pin Mapping: [PinMap2560 page](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the [Wire library](#). Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available [in the Arduino repository](#). The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160mil (0.16"), not an even multiple of the 100mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

ZOP Power 5500mAh Lipo Battery 11.1V 3S

.Chemistry: Li-Polymer

.Voltage:11.1V

.Capacity: 5500mAh

- . Discharge: 35C
- . Charge Plug: JST-XH
- . Discharge plug: XT60 Connector
- . Battery Dimension: 155mm x 45mm x 25mm (6.1x1.77x0.98inch)
- . Battery Weight: 417g / 14.7oz
- . Package Dimension: 165.00*60.00*35.00mm (6.49*2.36*1.38inch)
- . Package Weight: 440g / 15.5oz

ZOP Power 10000mAh Lipo Battery 11.1V 3S

- .Product type: Rechargeable lithium polymer battery
- .Voltage:11.1V
- .Configuration: 3S1P, 3 cells
- .Battery capacity: 10000MAh
- .Continuous discharge rate: 25C
- .Battery size: 162*65*28mm
- .Battery weight: 650G
- .Plug: XT60 Plug

3000mah Lipo Battery 11.1V 3S

- Battery Type: LiPo Battery
- Volts: 11.1v
- Cell Configuration: 3S1P
- Continuous Discharge: 35C (115.5A)
- Max Burst Rate: 70C (231A)
- Max Volts per Cell: 4.2V
- Max Volts per Pack: 12.6V
- Charge Rate: 1C
- Max Approved Charge Rate: 5C
- Wire Gauge: 12 AWG Soft and Flexible Low Resistance Silicone Wire
- Plug Type: Deans Plug/XT60
- Dimensions: 135 x 44 x 23 mm / 5.31 x 1.73 x 0.91 in (0 - 3mm difference)
- Weight: 9.88 oz (280g ± 2g)