



Σχολή  
Ηλεκτρολόγων  
Μηχανικών &  
Μηχανικών  
Υπολογιστών

# 2-axis Gimbal with AVR

Ενσωματωμένα Συστήματα Μικροεπεξεργαστών  
(ΗΡΥ411)

Τελική Αναφορά Project

Ομάδα:  
Καβαλλάρης Χρήστος  
Μάνεσης Αθανάσιος

# Περιεχόμενα

1. Εισαγωγή	3
1.1 Σκοπός του Project	3
1.2 Εξαρτήματα - Εργαλεία Ανάπτυξης	3
1.3 Στάδια / Milestones του Project	3
1.4 Παρόμοια Project	4
2. Θεωρία	4
2.1 Σταθεροποίηση Βάσης	4
2.2 PWM (Pulse Width Modulation)	5
2.3 Kalman και Complementary Φίλτρα	5
2.4 PID Controller	7
3. Τελική Σχεδίαση	7
3.1 Σταθεροποίηση βάσης χρησιμοποιώντας τις μετρήσεις.	7
3.2 Block Diagrams	8
3.3 Ανάλυση Κώδικα	8
3.4 Schematics	9
3.5 3D Model	10
3.6 Σύνοψη, Προβλήματα και Βελτιώσεις	10
4. Παράρτημα	12
4.1 Βιβλιογραφία και Πηγές	12
4.2 Κώδικας	12

# 1. Εισαγωγή

## 1.1 Σκοπός του Project

Σκοπός του Project ήταν η κατασκευή ενός 2-axis Gimbal, μιας συσκευής που θα μπορεί να σταθεροποιεί μια βάση, σε διάφορες περιστροφικές κινήσεις. Το gimbal που προσπαθήσαμε να υλοποιήσουμε, έχει δύο βαθμούς ελευθερίας δηλαδή θα έχει την δυνατότητα να αντισταθμίζει την περιστροφική κίνηση σε δύο άξονες. Επιλέξαμε να ασχοληθούμε με το Pitch και το Roll, τις κινήσεις γύρω από το x και y άξονα αντίστοιχα.

## 1.2 Εξαρτήματα - Εργαλεία Ανάπτυξης

Για την υλοποίηση του project χρησιμοποιήθηκαν τα παρακάτω εξαρτήματα:

- MPU6050: 6-axis Gyro & Accelerometer Sensor. Το output του μας δίνει την επιτάχυνση και την γωνιακή ταχύτητα.
- 2 x Fitec Continuous Servos (FS90R): Σερβοκινητήρες στους οποίους μπορούμε να ορίσουμε μόνο την ταχύτητα περιστροφής.
- ATmega16: 8-bit μικροελεγκτής της Atmel
- STK500

Η ανάπτυξη του κώδικα και όλες οι προσομοιώσεις έγιναν μέσω του AVR Studio 7.

Για την κατασκευή του gimbal επίσης εκτυπώθηκαν με 3D printer, η βάση, η λαβή και οι συνδέσεις που κάθονται οι servos. Η σχεδίαση τους έγινε μέσω της εφαρμογής Fusion 360.

## 1.3 Στάδια / Milestones του Project

Στα πλαίσια του μαθήματος το project χωρίστηκε σε τρία στάδια - milestones και το καθένα από αυτά είχε ένα ξεχωριστό στόχο. Στο πρώτο milestone εργαστήκαμε πάνω στον ATmega16 και στο πως να χρησιμοποιούμε τις δυνατότητες που μας προσφέρει. Γράψαμε κώδικα με στόχο να χρησιμοποιήσουμε interrupts, timers, watchdog timer, I/O και να βλέπουμε την συμπεριφορά που έχουμε ορίσει μέσω των LEDS της STK500. Στο δεύτερο milestone, ασχοληθήκαμε με το πως θα δημιουργήσουμε interfaces για τα εξαρτήματα και κυρίως για τον αισθητήρα ο οποίος αποτελεί την είσοδο του συστήματος μας και η δημιουργία interfaces ήταν πολυπλοκότερη. Χρησιμοποιώντας τα πρωτόκολλο I2C και USART καταφέραμε την επικοινωνία και το initialize του αισθητήρα και να δούμε στο Data Visualizer τις μετρήσεις που παίρνουμε σε πραγματικό χρόνο. Αντίθετα, ο servo δεν απαιτούσε την δημιουργία κάποιου interface αλλά μόνο μία τροφοδοσία (4.8V - 6V) και την χρήση του Timer1 σε PWM mode, για τον καθορισμό της κίνησης του. Τέλος, για το τρίτο milestone, στόχος ήταν η τελική υλοποίηση του gimbal με πλήρη λειτουργικότητα.

## 1.4 Παρόμοια Project

Αναζητήσαμε στο διαδίκτυο παρόμοια projects για την διευκόλυνση μας στον τρόπο που θα αντιμετωπίσουμε προβλήματα. Τα περισσότερα από αυτά, χρησιμοποιούσαν arduino μαζί με servos και MPU6050 ως κύριο αισθητήρα. Αυτό μας οδήγησε στην απόφαση να χρησιμοποιήσουμε και εμείς αυτά τα εξαρτήματα. Όταν τελικά κληθήκαμε να κάνουμε την τελική σχεδίαση του συστήματος, αντιληφθήκαμε πως μια υλοποίηση με τα συγκεκριμένα εξαρτήματα δεν θα μπορούσε να καλύψει απολύτως σωστά την λειτουργικότητα που είχαμε περιγράψει, αφού ο αισθητήρας έχει πολλές μεταβολές στις μετρήσεις του, αλλά και οι κινήσεις των servo δεν μπορούν να αντισταθμίσουν αμέσως την κίνηση που γίνεται από το χέρι μας, ώστε η βάση να ισορροπεί άμεσα, χωρίς κάποια καθυστέρηση.

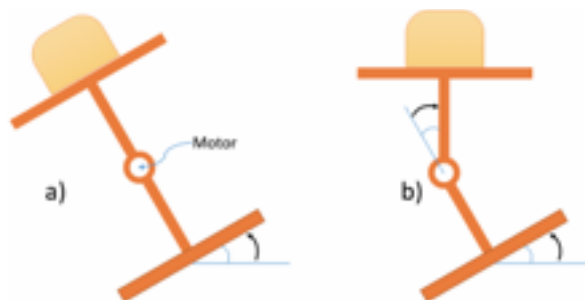
Επιπλέον, δεν βρήκαμε κάποιο project υλοποιημένο με AVR μικροελεγκτή, εκτός από ένα που χρησιμοποιούσε τον ATmega328p, τον μικροελεγκτή που χρησιμοποιούν τα περισσότερα Arduino, όπου και αυτό το project μπορούσε να χρησιμοποιεί έτοιμες βιβλιοθήκες της Arduino για την υλοποίηση των διεπαφών και την σχετική κίνηση των σερβοκινητήρων.

## 2. Θεωρία

Η ενότητα της θεωρίας παρέχει την απαραίτητη βασική γνώση για να γίνει κατανοητό το πρόβλημα καθώς και η λύση του.

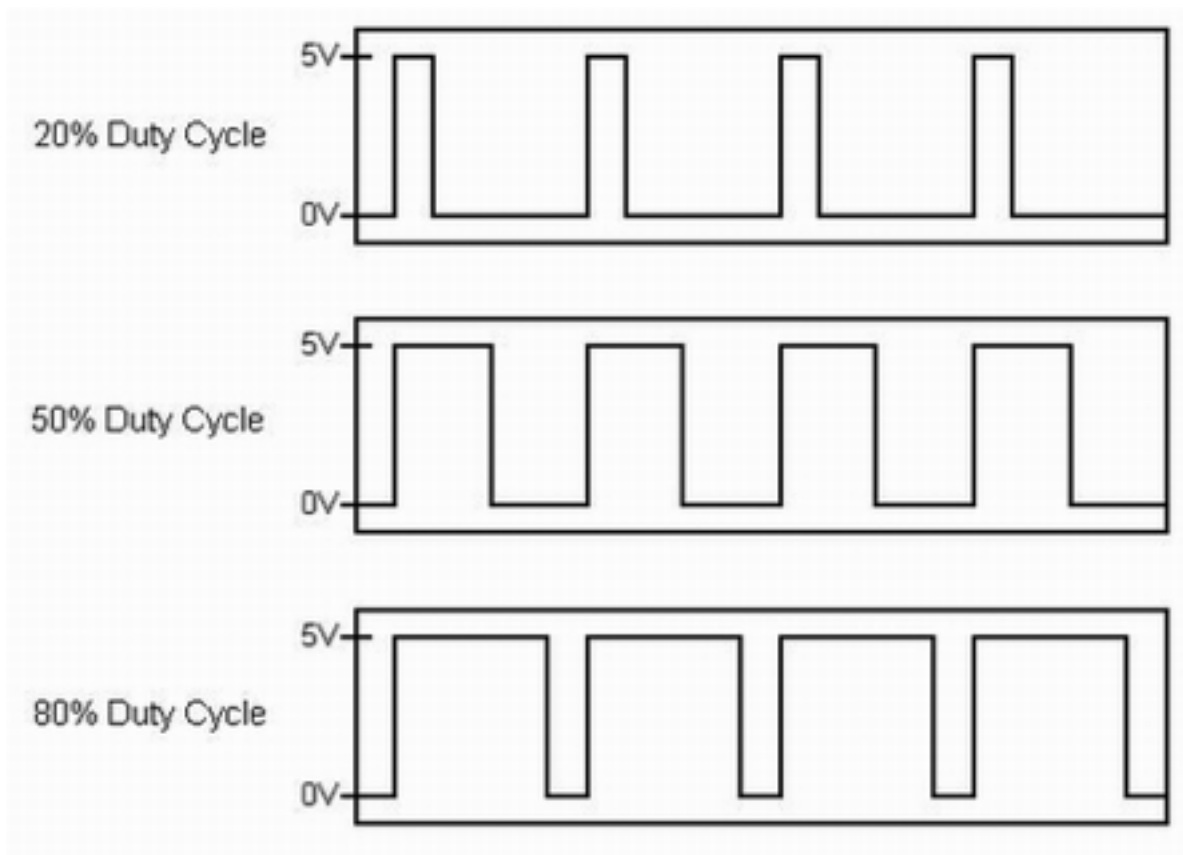
### 2.1 Σταθεροποίηση Βάσης

Ένα σύστημα σταθεροποίησης βάσης λαμβάνει συνήθως υπόψη το λιγότερο δύο γωνίες, για παράδειγμα το roll και το pitch που ασχοληθήκαμε εμείς. Σκοπός είναι να συντηρεί την βάση οριζοντιομένη. Αυτό μπορεί να γίνει χρησιμοποιώντας μια συσκευή η οποία μετράει την γωνία στην οποία βρίσκεται, όπως ένας γυροσκοπικός αισθητήρας ή ένα επιταχυνσιόμετρο.



## 2.2 PWM (Pulse Width Modulation)

Το σήμα PWM είναι ένας παλμος από μηδενικά και άσσους, που επιτρέπει στην τάση να είναι ένα ποσοστό της μέγιστης τάσης που μπορεί να περάσει από ένα κανάλι. Είναι συνήθως ανάμεσα στο διάστημα 0-255, όπου 0 αντιστοιχίζεται στο 0% της διαθέσιμης μέγιστης τάσης, ενώ το 255 στο 100%. Σε ένα παλμό, ένα σήμα του 127, αντιστοιχεί σε ένα σήμα που θα είναι 1 στην μέση της περιόδου του παλμού και 0 από την μέση μέχρι το τέλος του. Το σήμα αυτό μας είναι ιδιαίτερα χρήσιμο ώστε να ελέγξουμε την κίνηση των σερβοκινητήρων.



## 2.3 Kalman και Complementary Φίλτρα

Ο θόρυβος στις μετρήσεις που παίρνουμε από τον αισθητήρα μπορεί να βελτιωθεί με την χρήση ενός Kalman φίλτρου. Πρόκειται για έναν αναδρομικό αλγόριθμο, που απαιτεί μόνο την γνώση της προηγούμενης κατάστασης, δηλαδή της προηγούμενης μέτρησης, ώστε να κάνει πρόβλεψη της μιας πιο ρεαλιστικής μέτρησης.

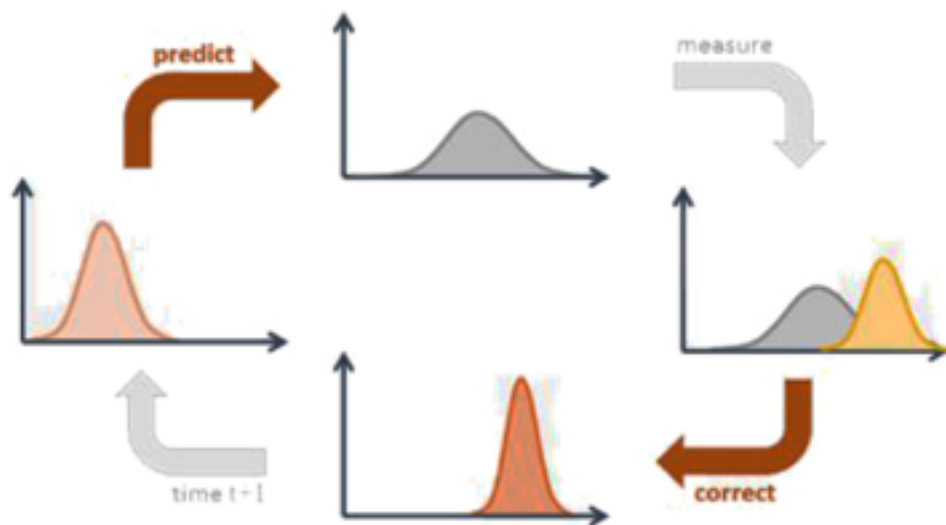


Figure 2.5: The Kalman cycle, Source:[Codeproject.com, 2016]

Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένα Complementary φίλτρο, το οποίο έχει πιο απλή λειτουργία. Γενικά ισχύει ότι οι μετρήσεις του επιταχυνσιόμετρου βραχυπρόθεσμα θα έχουν αρκετό θόρυβο, ενώ αντίθετα, οι μετρήσεις του γυροσκόπιου μακροπρόθεσμα θα έχουν σφάλματα, οπότε αν συνδιάσω τις μετρήσεις του επιταχυνσιόμετρου μακροπρόθεσμα και του γυροσκόπιου βράχυπρόθεσμα, παίρνω μια καλή εκτίμηση της γωνίας.

```
angle = (0.98) * (angle + gyro*dt) + (0.02) * (x_acc);
```

⏟
⏟  
 Integration.
 Low-pass portion acting on the  
accelerometer.

Something resembling a high-pass filter on the integrated gyro angle estimate. It will have approximately the same time constant as the low-pass filter.

## 2.4 PID Controller

Ένας PID ελεγκτής αποτελείται από τρία μέρη, το P (proportional), το I (integral) και το D (derivate) και αποτελεί ένα σύστημα ελέγχου ανάδραση. Έχει εφαρμογές σε συστήματα που μπορούν να μοντελοποιηθούν μαθηματικά και έχουν ένα προς ένα σχέση εισόδου-εξόδου. Όσον αφορά το Proportional κομμάτι, το σήμα σφάλματος είναι ανάλογο του σήματος ελέγχου. Το πρόβλημα με αυτό, είναι ότι αν μικρύνει το σφάλμα, μικραίνει και το σήμα ελέγχου και κατ' επέκταση δεν εξουδετερώνεται. Για να ελαχιστοποιήσουμε το σφάλμα, πρέπει η σταθερά P να αυξηθεί, το οποίο μειώνει την σταθερότητα του συστήματος.

Για το λόγο αυτό πρέπει να λάβουμε υπόψη το Integral κομμάτι. Το συγκεκριμένο θα καταχωρεί το σφάλμα μεταξύ του σήματος αναφοράς και των εξόδων και θα αυξάνει το σήμα εισόδου ανάλογα. Παρολ' αυτά μπορεί πάλι να μειωθεί η σταθερότητα του συστήματος, για αυτό χρειάζεται το Derivative κομμάτι.

Το Derivative κομμάτι κάνει το σύστημα πιο σταθερό με το να προβλέπει την μελλοντική κίνηση του συστήματος. Αυτό θέτει το σύστημα ευάλωτα στις διαταραχές, για αυτό και καθιστάται αναγκαίο το φιλτράρισμα σήματος.

Το PID αποτελεί το άθροισμα των τριών παραγόντων όπως φαίνεται από τον παρακάτω τύπο:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \left( \frac{d}{dt} e(t) \right)$$

## 3. Τελική Σχεδίαση

### 3.1 Σταθεροποίηση βάσης χρησιμοποιώντας τις μετρήσεις.

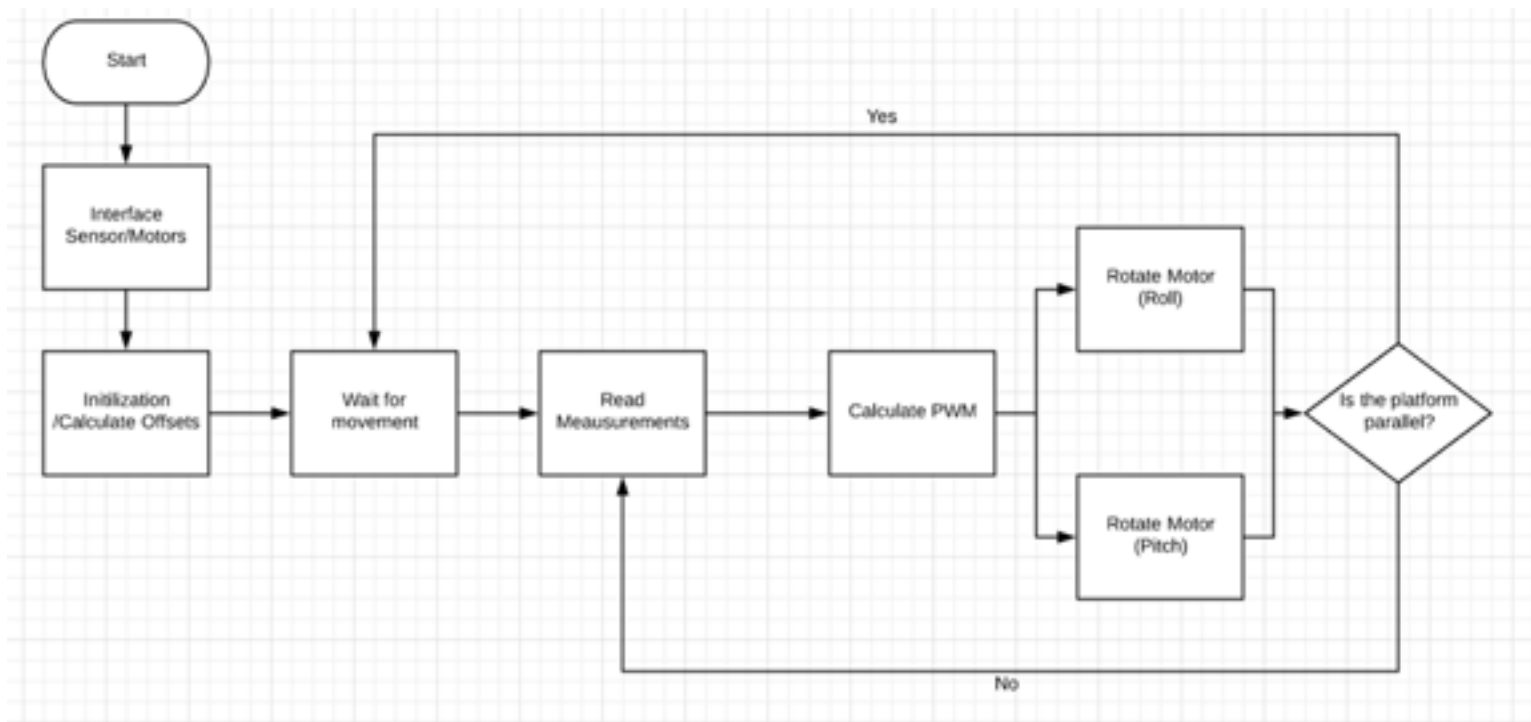
Η αρχική ιδέα για την τελική υλοποίηση ήταν να υπολογίσουμε την γωνία της κάθε κίνησης στους άξονες που αφορούν το pitch και το roll και ανάλογα να θέτουμε σε κίνηση τους σερβοκινητήρες σε αυτές τις γωνίες ώστε η βάση να παραμένει σταθερή. Λόγω του ότι χρησιμοποιήσαμε continuous servos, δεν ήταν δυνατό να θέτουμε γωνία κίνησης αλλά μόνο ταχύτητα περιστροφής. Οπότε, χρησιμοποιήσαμε τις μετρήσεις του επιταχυνσιόμετρου, αφού γνωρίζουμε ότι στην οριζόντια θέση η επιτάχυνση του άξονα z είναι ίση με την επιτάχυνση της βαρύτητας, δηλαδή 1g ενώ στον άξονα y είναι ίση με 0g. Μέσω loops, θέτουμε σε κίνηση τους σερβοκινητήρες όταν το σύστημα δεν είναι οριζόντιο μέχρι να φτάσουμε τις προαναφερθείσες τιμές στους δύο άξονες.

Terminal 7											
Ax = -0.15 g	Ay = -0.05 g	Az = 0.99 g	Gx = -35.18ψ/s	Gy = 0.44ψ/s	Gz = -0.56ψ/s						
Ax = -0.14 g	Ay = -0.06 g	Az = 1.00 g	Gx = -35.42ψ/s	Gy = 0.31ψ/s	Gz = -0.40ψ/s						
Ax = -0.14 g	Ay = -0.01 g	Az = 1.01 g	Gx = -35.42ψ/s	Gy = 0.36ψ/s	Gz = -0.81ψ/s						
Ax = -0.13 g	Ay = -0.02 g	Az = 1.02 g	Gx = -35.18ψ/s	Gy = 0.20ψ/s	Gz = -0.82ψ/s						
Ax = -0.13 g	Ay = -0.04 g	Az = 1.01 g	Gx = -35.42ψ/s	Gy = 0.22ψ/s	Gz = -0.44ψ/s						
Ax = -0.13 g	Ay = -0.04 g	Az = 1.00 g	Gx = -35.42ψ/s	Gy = 0.10ψ/s	Gz = -0.63ψ/s						
Ax = -0.11 g	Ay = -0.02 g	Az = 1.02 g	Gx = -35.42ψ/s	Gy = 0.09ψ/s	Gz = -0.56ψ/s						
Ax = -0.09 g	Ay = -0.02 g	Az = 1.00 g	Gx = -35.18ψ/s	Gy = 0.18ψ/s	Gz = -0.30ψ/s						
Ax = -0.10 g	Ay = -0.02 g	Az = 0.99 g	Gx = -35.18ψ/s	Gy = 0.09ψ/s	Gz = -0.92ψ/s						
Ax = -0.12 g	Ay = -0.04 g	Az = 0.99 g	Gx = -35.18ψ/s	Gy = 0.12ψ/s	Gz = -0.57ψ/s						
Ax = -0.10 g	Ay = -0.02 g	Az = 1.01 g	Gx = -35.18ψ/s	Gy = 0.40ψ/s	Gz = -0.50ψ/s						
Ax = -0.08 g	Ay = 0.00 g	Az = 1.02 g	Gx = -35.18ψ/s	Gy = 0.28ψ/s	Gz = -0.32ψ/s						

Η υλοποίηση παρουσιάζει αρκετά προβλήματα, αφού η σταθεροποίηση της βάσης δεν γίνεται άμεσα στις κινήσεις που θέτουμε το gimbal και κάποιες φορές αργεί μέχρι να κατασταλάξει στην οριζόντια θέση ή έχει απροσδιόριστη λειτουργία.

### 3.2 Block Diagrams

Στο παρακάτω block diagram μπορούμε να δούμε την top level συμπεριφορά του συστήματος.



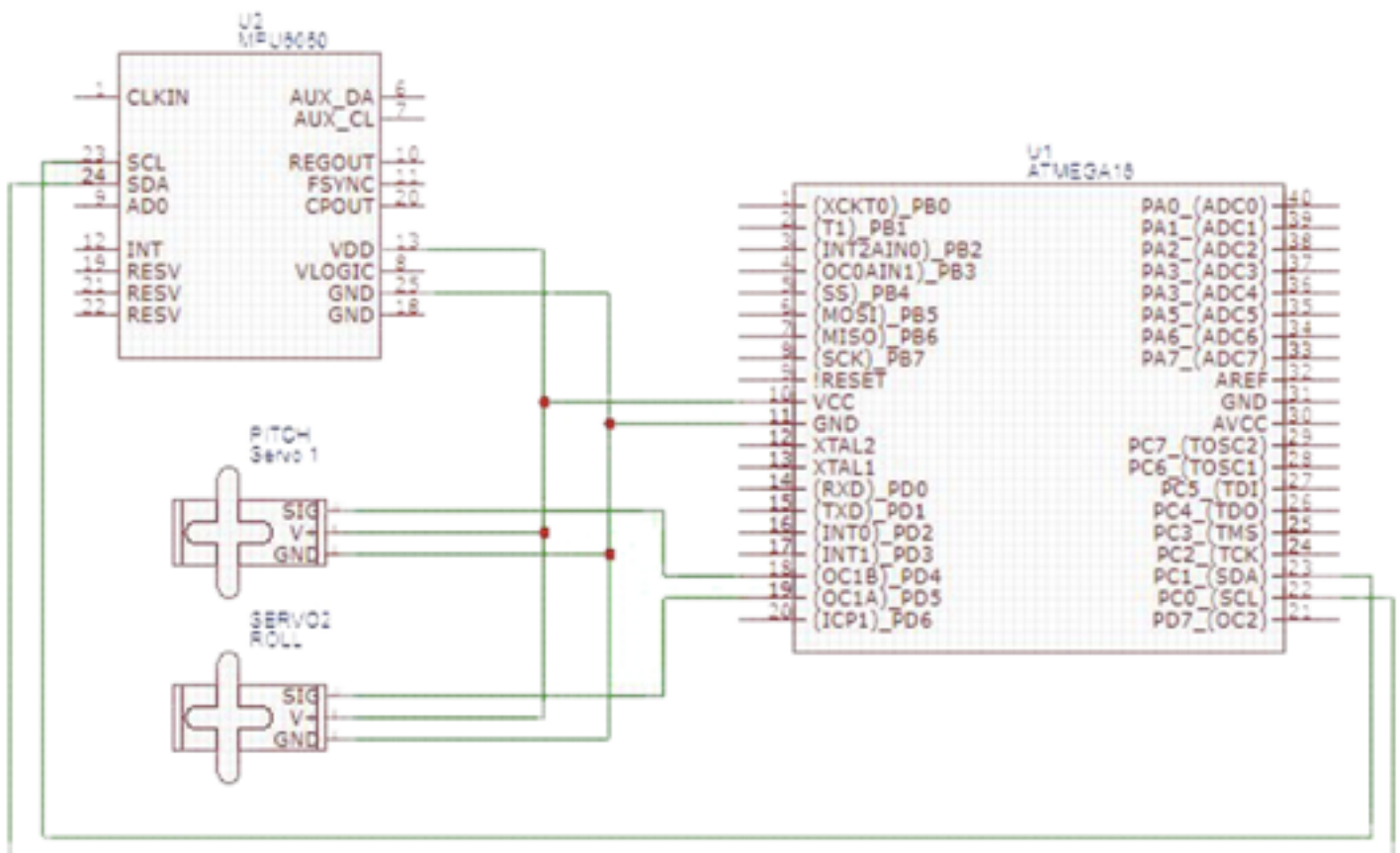
### 3.3 Ανάλυση Κώδικα

Πιο αναλυτικά, αφού αρχικοποιηθούν οι servos και ο αισθητήρας, καταγράφονται οι αρχικές συντεταγμένες της βάσης και καθορίζεται αυτή ως αρχική επιθυμητή θέση που θέλουμε να παραμείνει. Το σύστημα βρίσκεται σε αναμονή μέχρι η βάση να βρεθεί σε διαφορετική θέση, όπου και οι ενδείξεις του επιταχυνσιόμετρου αλλάζουν. Με βάση τις ενδείξεις, καθορίζουμε την φορά κίνησης των κινητήρων καθώς και την διάρκεια κίνησης τους, ενώ η ταχύτητά τους είναι σταθερή. Πειραματικά ορίσαμε κάποια εύρη μετρήσεων στα οποία αναθέσαμε καταστάσεις του συστήματος, λόγω ότι οι μετρήσεις δεν είναι σταθερές και αυτό θα προκαλούσε μεγάλη διαταραχή στην κίνηση. Η παραπάνω διαδικασία είναι επαναληπτική έως ώτου η πλατφόρμα βρεθεί στην αρχική της κατάσταση. Για την υλοποίηση των παραπάνω χρησιμοποιήθηκαν και οι δύο timers του μικροελεγκτή καθώς και polling για την κίνηση των κινητήρων.



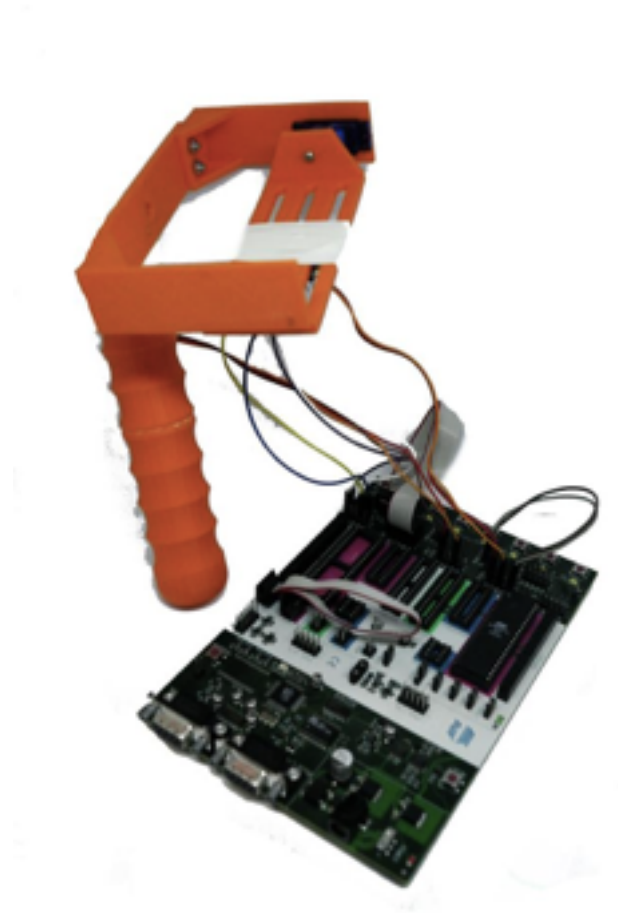
### 3.4 Schematics

Παρακάτω βλέπουμε την συνδεσμολογία του συστήματος. Η STK500 τροφοδοτείται με 12V, ενώ τα servos και ο MPU6050 με 5V από τα VTG της STK500.



### 3.5 3D Model

Η τελική κατασκευή του Gimbal.

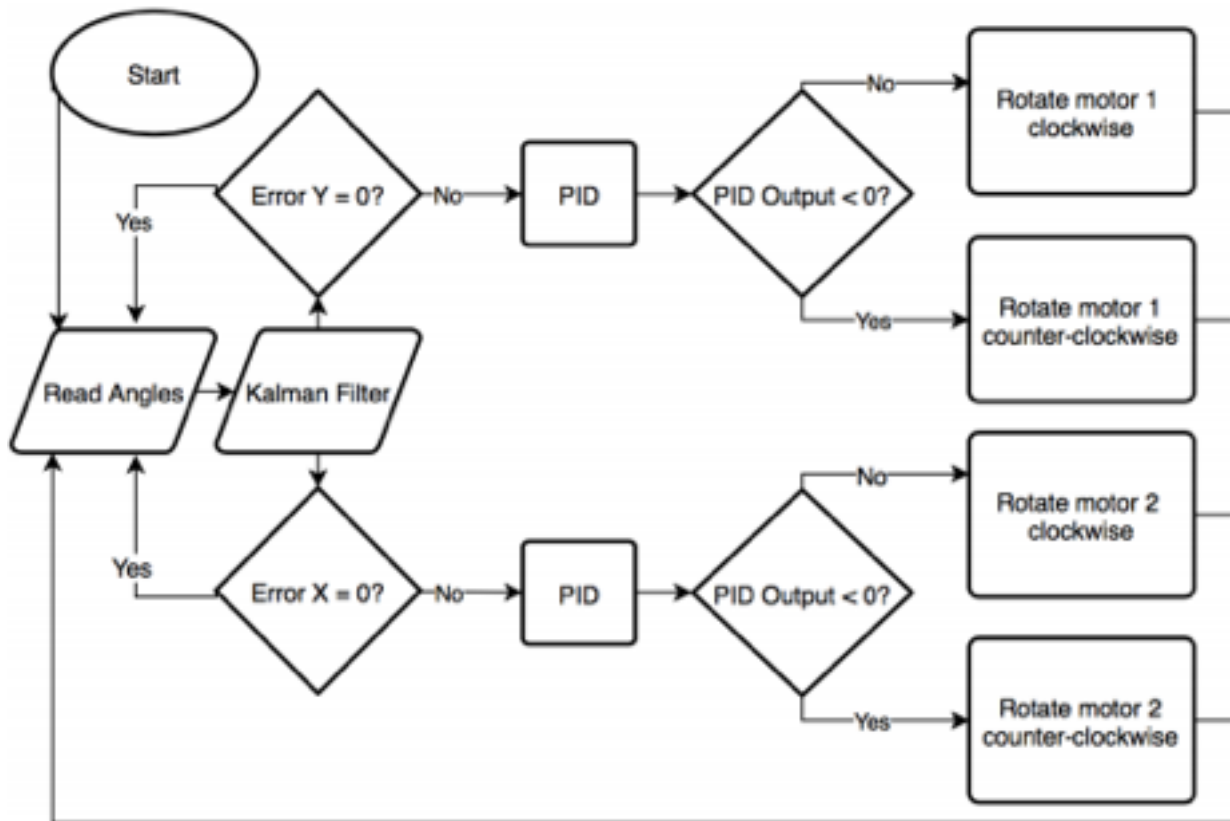


### 3.6 Σύνοψη, Προβλήματα και Βελτιώσεις

Η τελική υλοποίηση του συστήματος δεν είναι αυτή που επιθυμούσαμε. Αντιμετωπίσαμε πολλά προβλήματα, που κατά κύριο λόγο ευθύνεται το γεγονός ότι τα components που επιλέξαμε δεν ήταν κατάλληλα για την επιθυμητή λειτουργικότητα. Οι συγκεκριμένοι σερβοκινητήρες επειδή είναι συνεχόμενης κίνησης, μπορούν να μεταβάλλουν μόνο την επιτάχυνση κίνησης τους, για αυτό και γίνεται δύσκολη η αντιστάθμιση της κίνησης με βάση την γωνία περιστροφής της πλατφόρμας.

Εναλλακτικά, η δομή του κώδικα που θέλαμε να υλοποιήσουμε αλλά δεν καταφέραμε, περιγράφεται από το παρακάτω block diagram και περιέχει την θεωρητική ανάλυση που έχουμε κάνει, προσδιορίζοντας την γωνία που βρίσκεται η πλατφόρμα στους δύο άξονες ώστε να υπάρχει ανάλογη κίνηση στους κινητήρες, χρησιμοποιώντας PID controllers για το έλεγχο της φοράς κίνησης των κινητήρων, Kalman φίλτρο για την βελτίωση των μετρή-

σεων και brushless κινητήρες για την ομαλή κίνηση της πλατφόρμας και την άμεση αντιστάθμιση των μεταβολών της.



Τέλος, στο μέλλον θα θέλαμε να προσθέσουμε και τρίτο βαθμό ελευθερίας στο gimbal, μαζί με την υλοποίηση που περιγράφεται στο block diagram, ώστε να χει λειτουργικότητα παρόμοια με αυτή που έχουν εμπορικά πακέτα.

## 4. Παράρτημα

### 4.1 Βιβλιογραφία και Πηγές

- Atmel Community: <http://www.avrfreaks.net/>
- ATmega16 Manual: <http://www.atmel.com/images/doc2466.pdf>
- MPU6050 Manual: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- Servos Theory: <https://learn.sparkfun.com/tutorials/continuous-rotation-servo-trigger-hookup-guide>
- Kalman Filters Theory: [http://staff.ulsu.ru/semushin/\\_index/\\_pilocus/\\_gist/docs/mycourseware/13-stochmod/2-reading/grewal.pdf](http://staff.ulsu.ru/semushin/_index/_pilocus/_gist/docs/mycourseware/13-stochmod/2-reading/grewal.pdf)

### 4.2 Κώδικας

Οι κώδικες όλων των milestone καθώς και η τελική υλοποίηση βρίσκονται στο Github σε αντίστοιχα branches.

<https://github.com/amanesis/Embedded-Systems-Project>