



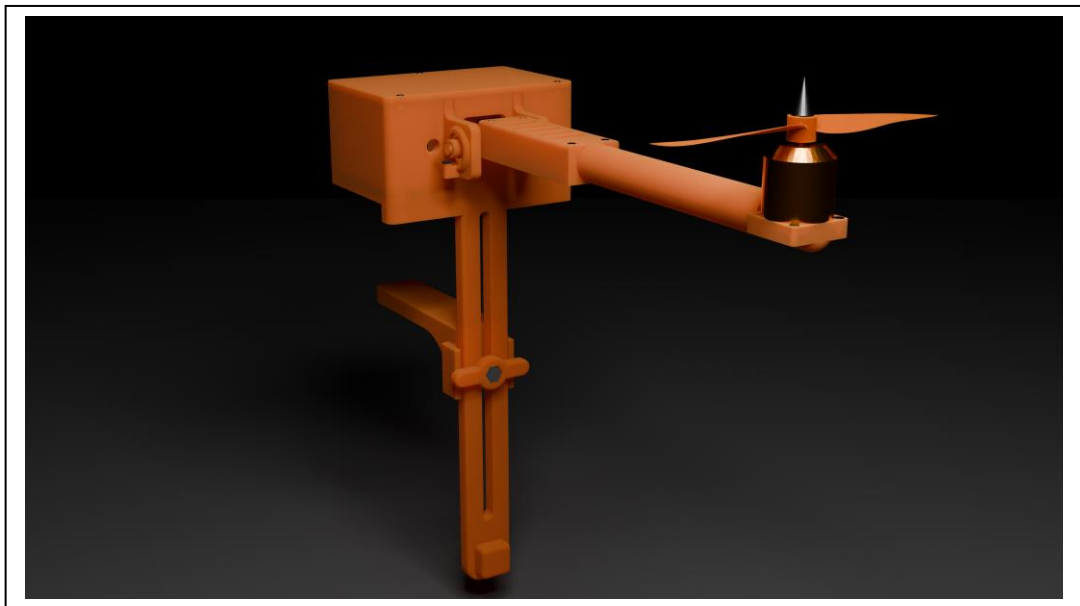
**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ**

## **Διπλωματική Εργασία**

**Ελικοφόρο εκκρεμές ως εκπαιδευτικό εργαλείο για PID ελεγκτές**



**Φοιτητής: Κουτουλάκης Γρηγόριος**  
**ΑΜ: eee18387224**

**Επιβλέπων Καθηγητής**

**Παναγιώτης Τσιάκας**  
**Λέκτορας**

**ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΣΕΠΤΕΜΒΡΙΟΣ 2024**



**UNIVERSITY OF WEST ATTICA**  
**FACULTY OF ENGINEERING**  
**DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING**

## **Diploma Thesis**

### **Aero pendulum as an educational tool for PID controllers**



**Student: Koutoulakis Grigorios**  
**Registration Number: eee18387224**

**Supervisor**

**Panagiotis Tsiakas**  
**Academic Lecturer**

**ATHENS-EGALEO, SEPTEMBER 2024**

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Παναγιώτης Τσιάκας, Λέκτορας	Κουλούρας Γρηγόριος, Αναπληρωτής Καθηγητής	Ζώης Ηλίας, Αναπληρωτής Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

**Copyright ©** Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Κουτουλάκης Γρηγόριος,  
Σεπτέμβριος, 2024**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

### **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Κουτουλάκης Γρηγόριος του Γεωργίου, με αριθμό μητρώου 18387224 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

**δηλώνω υπεύθυνα ότι:**

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.»

Ο Δηλών  
Κουτουλάκης Γρηγόριος



## Περίληψη

Στην παρούσα διπλωματική εργασία θα αναλυθεί το εκπαιδευτικό εργαλείο που κατασκευάστηκε. Πρόκειται για ένα πλήρες εκπαιδευτικό πακέτο, του οποίου στόχος είναι η εκμάθηση και εξοικείωση με τους PID ελεγκτές από άτομα που ενδιαφέρονται για το θέμα. Αποτελείται από ένα ηλεκτρομηχανικό σύστημα που συνοδεύεται από εφαρμογές για τον έλεγχο του. Σκοπός του, να βοηθήσει στην κατανόηση των PID ελεγκτών μέσα από πρακτικό παράδειγμα και επαφή με το ίδιο το σύστημα. Ο τρόπος διανομής του είναι σε ψηφιακή μορφή μέσω online αποθετηρίου το οποίο περιέχει όλα τα απαραίτητα αρχεία και εφαρμογές, για την κατασκευή του ή την τροποποίησή του από τον χρήστη, όπως και αναλυτικές οδηγίες και βίντεο λειτουργίας.

Το σύστημα διαθέτει έναν βραχίονα συνδεδεμένο σε μία σταθερή βάση μέσω άρθρωσης στο ένα άκρο του και ένα ηλεκτρικό κινητήρα και έλικα στο άλλο, κάνοντας έτσι μια απλή κίνηση, όπως ένα εκκρεμές, το οποίο με τη βοήθεια του έλικα μπορεί να σταθεροποιηθεί σε διαφορετικές γωνίες. Όλο το σύστημα συγκρατείται πάνω σε οποιοδήποτε πάχους γραφείο με ρυθμιζόμενη βάση και είναι εύκολο στην μεταφορά. Το μεγαλύτερο μέρος των μηχανικών εξαρτημάτων είναι εκτυπωμένα με τρισδιάστατο εκτυπωτή (3D printer). Περιέχει μικροελεγκτή βασισμένο στο ESP32 της Espressif, αισθητήρα για τον προσδιορισμό της γωνίας του βραχίονα και ελεγκτή ταχύτητας για τον ηλεκτρικό κινητήρα. Όλα τα ηλεκτρονικά είναι συνδεδεμένα σε τυπωμένη πλακέτα (PCB) που σχεδιάστηκε ειδικά για αυτόν τον σκοπό.

Συνοδεύεται από δύο εφαρμογές, μία για κινητές συσκευές (mobile) που τρέχουν λειτουργικό Android και μία για ηλεκτρονικό υπολογιστή (PC). Και οι δύο παρέχουν τα απαραίτητα για τον χειρισμό του συστήματος: τροποποιώντας τις τιμές  $K_p$ ,  $K_i$  και  $K_d$  του ελεγκτή, όπως και χειροκίνητο τρόπο λειτουργίας για αντιμετώπιση προβλημάτων (debugging). Ο τρόπος σύνδεσης της mobile εφαρμογής γίνεται ασύρματα με χρήση Bluetooth ενώ με την εφαρμογή υπολογιστή μέσω USB καλωδίου και σειριακής επικοινωνίας. Η εφαρμογή για PC διαθέτει και ψηφιακό τρισδιάστατο (3D) μοντέλο του συστήματος για ευκολότερη κατανόηση της λειτουργίας του. Η επικοινωνία του μικροελεγκτή με τις εφαρμογές γίνεται με κωδικοποίηση της πληροφορίας χρησιμοποιώντας πρωτότυπο πρωτόκολλο.

Στο online πακέτο περιλαμβάνονται όλα τα απαραίτητα αρχεία για την κατασκευή του συστήματος, όπως τα αρχεία για την 3D εκτύπωση, τα αρχεία για την εκτύπωση του PCB όπως και λίστα εξαρτημάτων με αναλυτικές λεπτομέρειες. Ακόμη, δίνονται τα εκτελέσιμα αρχεία των εφαρμογών όπως και ο πηγαίος κώδικάς τους και το λογισμικό του ESP32. Το πακέτο αυτό είναι άπειρα επεκτάσιμο χάρη στον ανοικτό κώδικα (open source) ώστε κάθε χρήστης να μπορεί να το προσαρμόσει στις δικές του προτιμήσεις. Η αλλαγή εξαρτημάτων, αισθητήρων όπως και συναρτήσεων, κώδικα και ελεγκτών είναι εφικτή, αν ο εκάστοτε χρήστης διαθέτει τις απαραίτητες γνώσεις, οι οποίες δεν απαιτούνται για το βασικό πακέτο.

## Λέξεις – κλειδιά

Εκπαιδευτικό Εργαλείο, PID ελεγκτής, Printed Circuit Board (PCB), 3D εκτύπωση, Unity, Mobile App, Προγραμματισμός Μικροελεγκτή

## **Abstract**

This thesis will describe the educational tool that was created. It is a full educational package, whose target is to teach interested people about PID controllers. It is an electromechanical system accompanied by applications to control it. Its purpose is to help the understanding of PID controllers through a practical example and constant interaction with the system itself. The distribution method is done digitally through an online repository which has all the required files and apps for the recreation or modification of the system depending on each user, as well as detailed instructions, documentation and video examples.

The system has an arm attached to a fixed base from a hinge and has an electric motor and a propeller on its other end, this way it makes a pendulum like movement and with the help of the propeller it can stabilize in different angles. The whole system is fixed to any desk thickness using an adjustable grip and it is very easy to move it around. Most of its mechanical parts are 3D printed using a printer. It contains a microcontroller based on Espressif's ESP32, a sensor to determine the angle of the arm and a speed controller for the electric motor. All electronic parts are connected in a custom-made PCB.

It is distributed along two apps, one for a mobile device running Android and one for a personal computer (PC). Both apps offer the required functions to control the system: by changing the  $K_p$ ,  $K_i$  and  $K_d$  parameters of the controller, as well as a manual mode for debugging. The mobile app is connected to the system wirelessly using Bluetooth and the PC version using serial communication. The PC version also has a 3D replica model of the system for the easier understanding of the functionality of the system. The communication with the microcontroller is accomplished by encoding the information using a prototype protocol specifically made for this purpose.

The online package has all the necessary files to construct the system, files for 3D printing, printing the PCB, as well as a list with all the parts used. Also, has all the executable files for both apps, their source code and the microcontroller firmware. This package is infinitely expandable because of the open-source nature of the code, so every user can adjust it based on their preferences. Changing parts, sensors and functions in the code can be easily done if the user has the appropriate knowledge, but this knowledge is not required for the use of the main package.

## **Keywords**

Educational Tool, PID controller, Printed Circuit Board (PCB), 3D printing, Unity, Mobile App, Microcontroller programming.

## Περιεχόμενα

<b>Κατάλογος Εικόνων .....</b>	<b>9</b>
<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>11</b>
<b>Αντικείμενο της διπλωματικής εργασίας .....</b>	<b>11</b>
<b>Σκοπός και στόχοι.....</b>	<b>11</b>
<b>Μεθοδολογία .....</b>	<b>11</b>
<b>Καινοτομία .....</b>	<b>12</b>
<b>Δομή</b>	<b>12</b>
<b>1 ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : Περιγραφή συστήματος και έρευνα.....</b>	<b>13</b>
<b>1.1 Θεωρητικό υπόβαθρο.....</b>	<b>13</b>
1.1.1 Εκπαιδευτικά συστήματα και αυτόματος έλεγχος .....	13
1.1.2 PID ελεγκτές .....	14
1.1.3 Ανοικτός κώδικας.....	16
<b>1.2 Το Ελικοφόρο Εκκρεμές ως εκπαιδευτικό εργαλείο για PID ελεγκτές.....</b>	<b>17</b>
<b>1.3 Επιλογή εξαρτημάτων.....</b>	<b>19</b>
1.3.1 Μικροελεγκτές .....	19
1.3.2 Ηλεκτρονικά.....	20
1.3.3 Αισθητήρας Κλίσης.....	22
1.3.4 Ηλεκτροκινητήρας .....	23
1.3.5 Μηχανικά μέρη .....	24
1.3.6 3D εκτύπωση.....	24
<b>1.4 Επιλογή προγραμμάτων.....</b>	<b>25</b>
<b>2 ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : Κατασκευή Συστήματος.....</b>	<b>27</b>
<b>2.1 3D Εκτύπωση.....</b>	<b>27</b>
2.1.1 Σχεδίαση στο FreeCAD .....	27
2.1.2 Εκτύπωση.....	29
<b>2.2 Πλακέτα.....</b>	<b>32</b>
2.2.1 Ηλεκτρονικό σχέδιο .....	32
2.2.2 Διάτρητη πλακέτα .....	33
2.2.3 Σχεδίαση στο KiCAD.....	34
2.2.4 Κατασκευή PCB.....	37
<b>2.3 Κατασκευή Συστήματος.....</b>	<b>38</b>
2.3.1 Κατασκευή Μηχανικού μέρους .....	38
2.3.2 Έλεγχος Ηλεκτρονικών .....	40
<b>3 ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> : Κώδικας Μικροελεγκτή .....</b>	<b>42</b>
<b>3.1 Arduino IDE .....</b>	<b>42</b>
<b>3.2 Είσοδος / Έξοδος Δεδομένων στον Μικροελεγκτή.....</b>	<b>42</b>
3.2.1 Είσοδος Αισθητήρα.....	42
3.2.2 Έξοδος στο Electronic Speed Controller .....	43
<b>3.3 Πρωτόκολλο Επικοινωνίας .....</b>	<b>44</b>
3.3.1 Bluetooth .....	44
3.3.2 UART.....	45
3.3.3 Αποστολή Δεδομένων .....	45
3.3.4 Λήψη Δεδομένων .....	46
<b>3.4 Συναρτήσεις .....</b>	<b>46</b>
3.4.1 Χειροκίνητη Λειτουργία .....	47
3.4.2 Λειτουργία PID .....	47

<b>4</b>	<b>ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> : Εφαρμογή Κινητού .....</b>	<b>49</b>
<b>4.1</b>	<b>Android Studio .....</b>	<b>49</b>
<b>4.2</b>	<b>Κώδικας Bluetooth.....</b>	<b>49</b>
<b>4.3</b>	<b>App User Interface .....</b>	<b>50</b>
<b>4.4</b>	<b>Ανάπτυξη.....</b>	<b>51</b>
4.4.1	Encode Decode Class .....	51
4.4.2	Βασική λογική .....	51
4.4.3	Χτίσιμο Εφαρμογής .....	51
<b>5</b>	<b>ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : Εφαρμογή Υπολογιστή .....</b>	<b>53</b>
<b>5.1</b>	<b>3D Μοντέλα στο Unity και Blender .....</b>	<b>54</b>
5.1.1	Μοντέλο Συστήματος.....	54
5.1.2	Μοντέλο Περιβάλλοντος Χώρου .....	55
<b>5.2</b>	<b>Σειριακή επικοινωνία.....</b>	<b>57</b>
<b>5.3</b>	<b>Unity User Interface.....</b>	<b>58</b>
5.3.1	COM port ρυθμίσεις.....	59
5.3.2	Ρυθμίσεις PID.....	59
5.3.3	Μπάρα πληροφοριών και ελέγχου .....	60
<b>5.4</b>	<b>Εφαρμογή σε Unity .....</b>	<b>61</b>
<b>6</b>	<b>ΚΕΦΑΛΑΙΟ 6<sup>ο</sup> : Διανομή Ανοικτού Κώδικα.....</b>	<b>62</b>
<b>6.1</b>	<b>Οδηγίες Χρήσης .....</b>	<b>62</b>
6.1.1	Γραπτές Οδηγίες.....	62
6.1.2	Οδηγίες σε Animation .....	62
6.1.3	Βίντεο Παρουσίασης Λειτουργιών .....	64
<b>6.2</b>	<b>Οργάνωση αρχείων σε Online Αποθετήριο.....</b>	<b>65</b>
	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>66</b>
	<b>Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές .....</b>	<b>68</b>
	<b>Παράρτημα Α.....</b>	<b>70</b>



## Κατάλογος Εικόνων

Εικόνα 1.1: Περιστροφικό Ανάστροφο εκκρεμές (1) (Rotary Inverted Pendulum, χ.χ.), Έλεγχος ταχύτητας DC κινητήρα (2) (Speed Control of a DC Motor, χ.χ.), Ελεγκτής για DC servomotor (3) (DELCOLABS, χ.χ.) .....	14
Εικόνα 1.2: Μπλοκ διάγραμμα ενός ελεγκτή σε σύστημα αυτομάτου ελέγχου (Block diagram of process control system, χ.χ.).....	15
Εικόνα 1.3: Μπλοκ διάγραμμα ενός PID ελεγκτή (PID Control, χ.χ.).....	16
Εικόνα 1.4: Σχεδιάγραμμα του συστήματος, σημειωμένες πάνω οι πιθανές θέσεις του βραχίονα.....	17
Εικόνα 1.5: Είσοδοι και έξοδοι του ESP32 DevKitC (ESP32-DevKitC V4 , n.d.) .....	20
Εικόνα 1.6: Διάτρητη πλακέτα πρωτοτύπων (1) (1528-5588-ND, n.d.), Screw terminal block (2) (Screw Terminal 2P 5mm, n.d.), Voltage Regulator (3) (LM1086 1.5-A, n.d.), Female pin header (4) (Pin Header 1x20 Female 2.54mm, n.d.), T plug (5) (T-Connector Plug, n.d.), Καλώδια με ακροδέκτη (6) (Wires 3-Pin 30cm Female to Female, n.d.) .....	21
Εικόνα 1.7: Σχεδιασμένο το Hall Effect, με βάση αυτό λειτουργεί ένας hall effect αισθητήρας (Hall Effect, χ.χ.).....	22
Εικόνα 1.8: Διαμετρικός μαγνήτης που χρησιμοποιήθηκε στην διάταξη (ΜΑΓΝΗΤΗΣ ΔΙΣΚΟΣ Φ6×2,5mm ΔΙΑΜΕΤΡΙΚΟΣ, n.d.).....	23
Εικόνα 1.9: Ο DC brushless κινητήρας A2212 (XD A2212 1400KV Brushless Motor, χ.χ.).....	23
Εικόνα 1.10: Το ρουλεμάν που τοποθετήθηκε στο σύστημα.....	24
Εικόνα 1.11: Ο 3D εκτυπωτής Creality CR-10 (3D printer Creality CR-10 v2 300, n.d.) .....	25
Εικόνα 2.1: Μερικά από τα μοντέλα των εξαρτημάτων που σχεδιάστηκαν στο FreeCAD .....	28
Εικόνα 2.2: Παραδείγματα σύνδεσης, με βίδες και threaded inserts (επάνω), με εγκοπές (κάτω) .....	28
Εικόνα 2.3: Παράμετροι που επιλέχθηκαν για τον εκτυπωτή, την εκτύπωση και το υλικό στο Cura .....	30
Εικόνα 2.4: Εξάρτημα στο Cura, με τον ενδεικτικό χρόνο εκτύπωσης και ποσότητα υλικού που θα χρησιμοποιηθεί κάτω δεξιά.....	31
Εικόνα 2.5: Εξαρτήματα με τα supports τους μόλις ολοκληρώθηκε η εκτύπωση .....	31
Εικόνα 2.6: Η βάση στήριξης και το ολοκληρωμένο σύστημα.....	32
Εικόνα 2.7: Το κύκλωμα του συστήματος, σχεδιασμένο στο KiCAD.....	33
Εικόνα 2.8: Το κύκλωμα σε διάτρητη πλακέτα πρωτοτύπων εμπρός και πίσω.....	34
Εικόνα 2.9: Το layout των εξαρτημάτων επάνω στο PCB .....	35
Εικόνα 2.10: Τα σχεδιασμένα traces επάνω στο PCB όπου ολοκληρώνουν τις συνδέσεις των εξαρτημάτων.....	36
Εικόνα 2.11: Το ολοκληρωμένο PCB (αριστερά), 3D απεικόνιση της τελικής πλακέτας με τα εξαρτήματά της (δεξιά) .....	36

Εικόνα 2.12: Η φόρμα ρυθμίσεων για το PCB από την εταιρία JLCPCB .....	37
Εικόνα 2.13: Το PCB έτοιμο με όλα τα εξαρτήματα, για να τοποθετηθεί στο σύστημα .....	38
Εικόνα 2.14: Το σύστημα ολοκληρωμένο και στερεωμένο σε γραφείο .....	39
Εικόνα 2.15: Τα σπασμένα μέρη μετά την πτώση, στα σημεία όπου υπήρχαν βίδες .....	40
Εικόνα 4.1: Τα UI view της εφαρμογής που αναπτύχθηκε. Η επιλογή συσκευής (1), το κύριο μενού (2), το view σε χειροκίνητη λειτουργία (3), το view σε λειτουργία PID .....	50
Εικόνα 5.1: Το περιβάλλον του προγράμματος Unity .....	53
Εικόνα 5.2: 3D αναπαράσταση του συστήματος στο περιβάλλον του Blender, με την ιεραρχία των μοντέλων να φαίνεται δεξιά.....	54
Εικόνα 5.3: Τα textures και το μοντέλο του συστήματος .....	55
Εικόνα 5.4: Το 3D μοντέλο του συστήματος με τα textures στα σωστά αντικείμενα .....	55
Εικόνα 5.5: Το μοντέλου γραφείου και περιβάλλοντος χώρου στο Blender .....	56
Εικόνα 5.6: Το εικονικό δωμάτιο με την κάμερα, την πηγή φωτισμού και το σύστημα στερεωμένο στο γραφείο.....	57
Εικόνα 5.7: Αλλαγή ρύθμισης του API Compatibility Level από τα Project Settings.....	57
Εικόνα 5.8: Η οθόνη των ρυθμίσεων, το πάνελ σύνδεσης επάνω δεξιά με μπλε χρώμα, το dropdown επιλογής λειτουργίας επάνω αριστερά και η μπάρα πληροφοριών στο κάτω μέρος .....	58
Εικόνα 5.9: Το πάνελ σύνδεσης COM port.....	59
Εικόνα 5.10: Οι ρυθμίσεις για την PID λειτουργία.....	60
Εικόνα 5.11: Η μπάρα στις τρεις καταστάσεις του συστήματος. Επάνω στην κατάσταση Idle, στη μέση στην χειροκίνητη λειτουργία με το slider και το κουμπί εφαρμογής και κάτω στην PID λειτουργία με την προβολή των παραμέτρων και τις γωνίες σταθεροποίησης.....	60
Εικόνα 5.12: Τα αρχεία που δημιουργήθηκαν μετά το Build της εφαρμογής .....	61
Εικόνα 6.1: Περιγραφή ενός keyframe, αριστερά ορίζεται η αρχική θέση του βραχίονα στον χώρο σε συγκεκριμένο χρόνο, δεξιά η τελική του θέση στον χώρο και χρόνο .....	63
Εικόνα 6.2: Μερικές εικόνες από το τελικό βίντεο animation.....	64
Εικόνα 6.3: Αποκόμματα κατά την επεξεργασία των βίντεο με το Davinci Resolve. Αριστερά η χρήση της mobile εφαρμογής, δεξιά χρήση εφαρμογής στο Unity. ....	64
Εικόνα 6.4: Ο φάκελος του πακέτου με όλα τα αρχεία οργανωμένα σε υποφακέλους .....	65

## ΕΙΣΑΓΩΓΗ

Οι ελεγκτές στα συστήματα αυτομάτου ελέγχου βρίσκονται γύρω μας σε επίπεδο καθημερινότητας, μαθαίνοντας για τους PID ελεγκτές, πολλοί μένουν μόνο στο θεωρητικό κομμάτι τους, χωρίς να εφαρμόσουν κάπου τις γνώσεις τους, μέχρι αυτό να γίνει σε κάποια σημαντική εργασία όπου τα περιθώρια λάθους είναι μικρά. Το γεγονός αυτό ώθησε την δημιουργία του πακέτου όπου καλείται να καλύψει αυτό το κενό, παρουσιάζοντας σε νέους μαθητές ένα σύστημα το οποίο τους προτρέπει στον πειραματισμό χωρίς κάποια σημαντική επίπτωση. Βασικό χαρακτηριστικό του είναι η ευκολία στην δημιουργία του από άτομα με περιορισμένες γνώσεις και η εύκολη εφαρμογή των PID ελεγκτών σε ένα σύστημα ειδικά σχεδιασμένο για αυτόν τον σκοπό.

Σκοπός του χρήστη είναι να βρει τις κατάλληλες τιμές που ορίζουν την συμπεριφορά του ελεγκτή με απλό και κατανοητό τρόπο, ώστε να πετύχουν σταθεροποίηση του συστήματος σε μία προκαθορισμένη από τους ίδιους θέση. Η χρήση του συστήματος γίνεται από δύο ειδικά σχεδιασμένες εφαρμογές και με πολύ απλά βήματα επιτρέπουν στον καθένα εύκολα και γρήγορα να τροποποιεί τις τιμές αυτές και να παρατηρεί το πως αντιδρά το σύστημα την επόμενη στιγμή.

Η κατασκευή του συστήματος και η επιλογή των υλικών έχει γίνει με γνώμονα τις πολλαπλές αποτυχίες που μπορεί να προκύψουν, ώστε το σύστημα να αντέξει μηχανικά οποιαδήποτε καταπόνηση από τις προσπάθειες σταθεροποίησης. Οι εφαρμογές όπως και το ίδιο το λογισμικό του μικροελεγκτή έχουν δημιουργηθεί με στόχο την αξιοπιστία και την σταθερότητα τους σε πολλαπλές περιπτώσεις, όπως και εμφάνιση μηνυμάτων για προστασία από λάθη που μπορεί να γίνουν εξαιτίας του κάθε χρήστη. Για την αναπαραγωγή του εργαλείου και την χρήση του από άλλους δεν απαιτείται ειδική γνώση μιας και υπάρχουν αναλυτικές οδηγίες που μπορεί να ακολουθήσει ο καθένας.

### Αντικείμενο της διπλωματικής εργασίας

Το κύριο θέμα της εργασίας είναι η περιγραφή της διαδικασίας που ακολουθήθηκε για την δημιουργία του προαναφερθέντος εκπαιδευτικού πακέτου. Αρχικά ποιοι είναι οι στόχοι του και πως αυτοί θα επιτευχθούν, θα παρουσιαστούν όλες οι επιλογές που πάρθηκαν κατά την κατασκευή του συστήματος και όλα τα προβλήματα και οι δυσκολίες που εμφανίστηκαν. Το πόσο σημαντικό είναι για άτομα που θέλουν να ασχοληθούν με το θέμα του αυτομάτου ελέγχου, όπως και με τον σχεδιασμό και την κατασκευή ηλεκτρονικών και μηχανικών συστημάτων ξεκινώντας μόνο από μια ιδέα. Τέλος την μετάδοση της εμπειρίας που αποκτήθηκε με την διαδικασία της εργασίας αυτής σε όλους τους ενδιαφερόμενους.

### Σκοπός και στόχοι

Στόχος της εργασίας αυτής είναι η δημιουργία και η διανομή ενός συστήματος το οποίο να είναι εύκολο στη χρήση χωρίς να απαιτείται μεγάλη γνώση σε κάτι αντίστοιχο, με σκοπό την εκμάθηση του χρήστη τόσο με το ίδιο το σύστημα όσο και με τους τρόπους ελέγχου που υπάρχουν. Όλα σε ένα ολοκληρωμένο πακέτο με αναλυτική τεκμηρίωση (documentation).

### Μεθοδολογία

Καθώς πρόκειται για κατασκευαστική εργασία με πραγματικό σύστημα, το οποίο μπορεί παρουσιαστεί και να χρησιμοποιηθεί, έγινε εκτενής μελέτη γύρω από παρόμοια συστήματα και

εξετάστηκαν πιθανά υλικά, εξαρτήματα και προγράμματα που χρησιμοποιήθηκαν. Μετέπειτα, έγινε η συλλογή αυτών από αντίστοιχα καταστήματα, ακολούθησε η κατασκευή του συστήματος και αργότερα έγινε ο προγραμματισμός τόσο του ίδιου του μικροελεγκτή όσο και των εφαρμογών που συνοδεύουν το σύστημα. Τέλος μαζεύτηκε όλο το υλικό που χρησιμοποιήθηκε ή δημιουργήθηκε σε ένα σύστημα αρχείων έτοιμο για ανάρτηση σε αποθετήριο.

## **Καινοτομία**

Πρόκειται για πρωτότυπο σύστημα, το οποίο είναι επεκτάσιμο και τροποποιήσιμο ανάλογα τον κάθε χρήστη. Ταυτόχρονα είναι πολύ εύκολο στην χρήση και στην κατασκευή, κρατώντας το κόστος όσο πιο χαμηλά γίνεται. Βασικό χαρακτηριστικό δεν είναι μόνο η σωστή λειτουργία του συστήματος αλλά και η εκτενής καθοδήγηση των χρηστών μέσα από οδηγίες και έγγραφα για την κατανόηση της λειτουργίας του.

## **Δομή**

Στο 1<sup>ο</sup> Κεφάλαιο παρουσιάζεται όλο το θεωρητικό υπόβαθρο στο οποίο στηρίζεται το σύστημα. Ακολουθεί η κατασκευή του συστήματος με εκτενή ανάλυση στο 2<sup>ο</sup> Κεφάλαιο, στο 3<sup>ο</sup> Κεφάλαιο αναλύεται ο κώδικας του μικροελεγκτή. Στο 4<sup>ο</sup> και 5<sup>ο</sup> Κεφάλαιο περιγράφονται οι εφαρμογές που δημιουργήθηκαν. Στο τελευταίο 6<sup>ο</sup> Κεφάλαιο γίνεται αναφορά στον τρόπο οργάνωσης της πληροφορίας σε κατανοητά κομμάτια για τους πιθανούς χρήστες. Ακολουθούν παρατηρήσεις, συμπεράσματα και βελτιώσεις.

## 1 ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : Περιγραφή συστήματος και έρευνα

Στο κεφάλαιο αυτό θα γίνει αναφορά στο θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση του συστήματος και τους PID ελεγκτές. Θα γίνει επί μέρους ανάλυση των υλικών που απαιτούνται για την κατασκευή του συστήματος, τι ρόλο παίζουν στο σύστημα και έρευνα που έγινε γύρω από αυτά για την επιλογή του κατάλληλου. Ακόμη έρευνα έχει γίνει και για όλα τα προγράμματα που θα χρησιμοποιηθούν, τι απαιτήσεις ζητούνται, καθώς και παρόμοια που υπάρχουν.

### 1.1 Θεωρητικό υπόβαθρο

Βασικό κομμάτι της εργασίας αυτής είναι τα εκπαιδευτικά εργαλεία. Θα απαντηθεί το τι είναι ένα εκπαιδευτικό εργαλείο και τι μπορεί να προσφέρει στο κατάλληλο κοινό, όπως και το πόσο σημαντικό είναι για αυτό. Έχοντας επιλέξει τον PID ως τον βασικό ελεγκτή του συστήματος, αφού είναι αρκετά διαδεδομένος και εύκολος στην εφαρμογή, θα γίνει ανάλυσή του, θα παρουσιαστεί το μαθηματικό μοντέλο του και θα τονιστεί η σημαντικότητά του και η παρουσία του στην καθημερινότητά μας. Το εκπαιδευτικό εργαλείο που κατασκευάστηκε είναι πλήρως ανοικτό προς όλους για να το χρησιμοποιήσουν, τροποποιήσουν ή και να αναδημιουργήσουν. Θα δοθούν έννοιες του ανοικτού κώδικα, της σημασίας του και παραδείγματα στο πως ο ανοικτός κώδικας μπορεί να βελτιώσει την ποιότητα των προγραμμάτων και συστημάτων που υπάρχουν.

#### 1.1.1 Εκπαιδευτικά συστήματα και αυτόματος έλεγχος

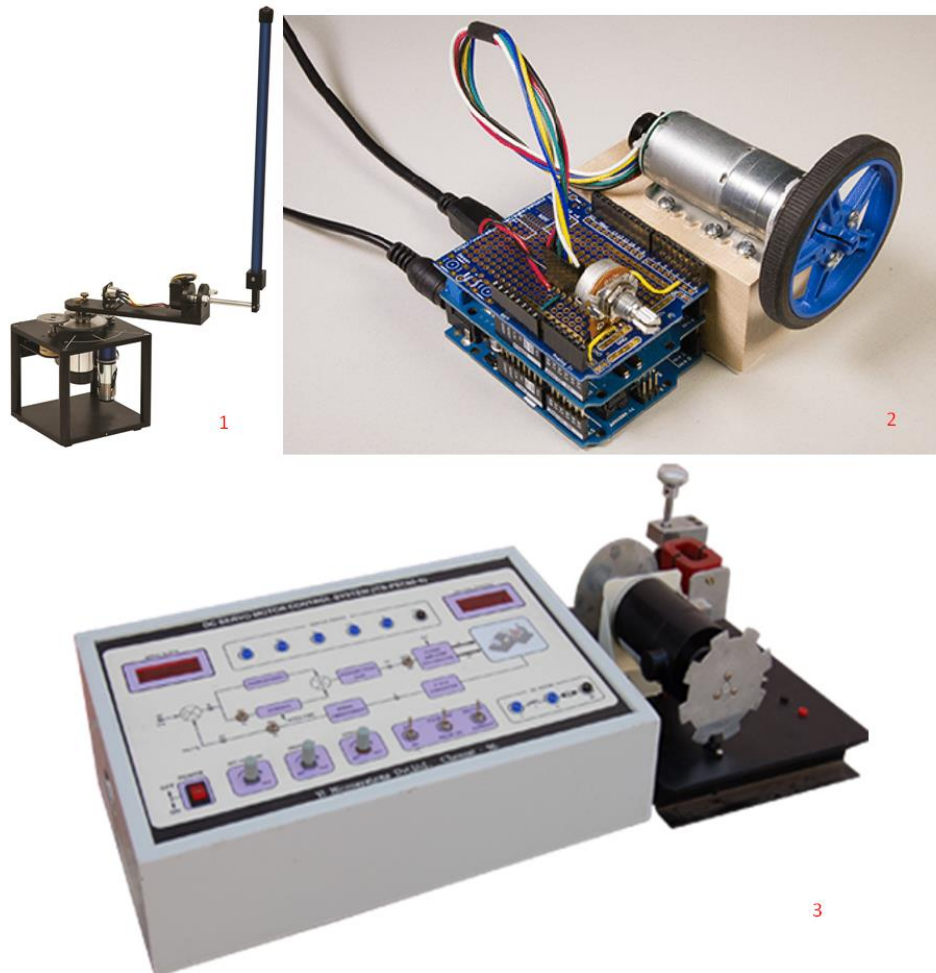
Ένα εκπαιδευτικό εργαλείο είναι οποιοσδήποτε πόρος πληροφορίας, συσκευή ή σύστημα το οποίο στοχεύει στην εκμάθηση ή εκπαίδευση. Τέτοια εργαλεία μπορούν να πάρουν διάφορες μορφές, από φυσικά αντικείμενα όπως εργαστηριακός εξοπλισμός, βιβλία και μοντέλα, μέχρι ψηφιακά μέσα όπως εφαρμογές, λογισμικά και online πλατφόρμες. Χρησιμοποιούνται ανά τον κόσμο από σχολεία, πανεπιστήμια και γενικά από άτομα με ενδιαφέρον για μάθηση. (Educational technology, χ.χ.)

Ιδιαίτερο ενδιαφέρον έχουν τα εργαλεία αυτά τα οποία έχουν μορφή φυσικού συστήματος. Τέτοια εργαλεία παίζουν σημαντικό ρόλο σε επίπεδο πανεπιστημίου ειδικά σε τομείς όπου η πρακτική εμπειρία είναι σημαντική. Τα συστήματα αυτά συμπεριλαμβάνουν εργαστηριακό εξοπλισμό, μοντέλα μηχανικών συστημάτων και γενικά οτιδήποτε είναι απτό και μπορεί να βοηθήσει στην εκμάθηση μέσα από αλληλεπίδραση με αυτό. Παράδειγμα τέτοιων εργαλείων είναι οι ρομποτικοί βραχίονες, ηλεκτρονικές πλακέτες ή διατάξεις ηλεκτρονικών εξαρτημάτων, συστήματα εφαρμογής αυτομάτου ελέγχου τα οποία μαθαίνουν σε ενδιαφερόμενους αρχές ρομποτικής, ηλεκτρικά φαινόμενα και αυτόματο έλεγχο αντίστοιχα. (ANDRONICEANU & BURLACU)

Ο αυτόματος έλεγχος είναι τομέας της μηχανικής, στον οποίο σχεδιάζονται συστήματα τα οποία ρυθμίζουν τις λειτουργίες τους, με βάση τις αλλαγές που γίνονται στο περιβάλλον τους, χωρίς την ανθρώπινη παρέμβαση. Τα συστήματα αυτά περιλαμβάνουν φαινόμενα ανάδρασης τα οποία ελέγχουν την κατάσταση του συστήματος και συμβάλουν στην προσαρμογή των συστημάτων για να πετύχουν κάποιο επιθυμητό αποτέλεσμα. Στόχος τους είναι η διατήρηση μιας σταθερής κατάστασης ανεξαρτήτως εξωτερικών αλλαγών και αναταραχών π.χ. η ρύθμιση της θερμοκρασίας ενός δωματίου, η ταχύτητα ενός αυτοκινήτου, η θέση κάποιου βραχίονα στο χώρο κ.λπ.

Μερικά εκπαιδευτικά εργαλεία ή διατάξεις που χρησιμοποιούνται στον αυτόματο έλεγχο είναι το ανάστροφο εκκρεμές, ο έλεγχος ταχύτητας DC κινητήρα και ο σέρβο-κινητήρας (servomotor). Το ανάστροφο εκκρεμές στοχεύει στην σταθεροποίηση ενός φαινομενικά ασταθούς συστήματος και

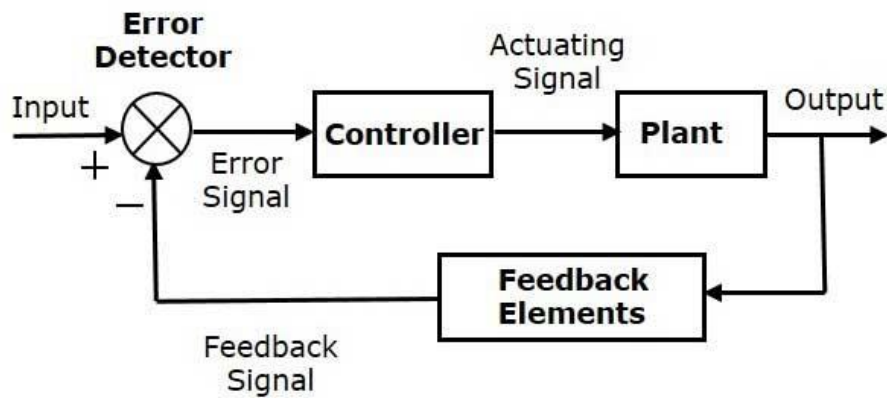
στοχεύει στην εκμάθηση της αναπαράστασης του συστήματος στον χώρο κατάστασης (state-space representation) και στους ελεγκτές πραγματικού χρόνου (real time control systems). Ο έλεγχος ταχύτητας DC κινητήρα διδάσκει τη χρήση ελεγκτών όπως ο PID για την ρύθμιση της ταχύτητας ή ροπής ενός κινητήρα. Τέλος, ο έλεγχος ενός servomotor έχει στόχο την σταθεροποίηση συστημάτων τα οποία απαιτούν μεγάλη ακρίβεια θέσης σε περιστροφικά ή γραμμικά συστήματα με χρήση ανάδρασης. (Κυο, 1987)



**Εικόνα 1.1: Περιστροφικό Ανάστροφο εκκρεμές (1) (Rotary Inverted Pendulum, χ.χ.), Έλεγχος ταχύτητας DC κινητήρα (2) (Speed Control of a DC Motor, χ.χ.), Ελεγκτής για DC servomotor (3) (DELCOLABS, χ.χ.)**

### 1.1.2 PID ελεγκτές

Οι ελεγκτές στα συστήματα αυτομάτου ελέγχου είναι συσκευές ή αλγόριθμοι οι οποίοι τροποποιούν και ρυθμίζουν ένα σύστημα ή διαδικασία. Στόχος τους η σταθεροποίηση της εξόδου του συστήματος επηρεάζοντας την είσοδό του με βάση την ανάδραση που λαμβάνουν. Κατά την λειτουργία τους, οι ελεγκτές λαμβάνουν την τρέχουσα κατάσταση του συστήματος και την συγκρίνουν με την επιθυμητή υπολογίζοντας την διαφορά τους, το σφάλμα. Με το σφάλμα ο ελεγκτής υπολογίζει το πώς θα προσαρμόσει την είσοδο στο σύστημα ώστε να πετύχει την επιθυμητή έξοδο.



**Εικόνα 1.2: Μπλοκ διάγραμμα ενός ελεγκτή σε σύστημα αυτομάτου ελέγχου (Block diagram of process control system, χ.χ.)**

Ο PID ελεγκτής υπολογίζει συνεχώς το σφάλμα (error) που προκύπτει από την διαφορά της τρέχουσας τιμής εξόδου του συστήματος με την επιθυμητή. Προσαρμόζει την είσοδο του συστήματος έτσι ώστε να ελαχιστοποιήσει το σφάλμα αυτό και ιδανικά να το μηδενίσει. Συνδυάζει τρία βασικά κομμάτια: το αναλογικό κομμάτι (proportional) (P), το ολοκληρωτικό κομμάτι (integral) (I) και το παραγωγικό (derivative) (D) για να δημιουργήσει το κατάλληλο σήμα εισόδου. Παρακάτω επεξηγείται κάθε κομμάτι ξεχωριστά.

**Proportional (P):** Ο proportional όρος είναι άμεσα ανάλογος του σφάλματος. Αν το σφάλμα είναι μεγάλο, τότε θα γίνει και μεγάλη διόρθωση. Η αναλογική ενίσχυση (proportional gain) ( $K_p$ ) καθορίζει το πόσο έντονη θα είναι αυτή η διόρθωση. Όταν χρησιμοποιείται μόνο του είναι πιθανό να δημιουργήσει σφάλμα σταθερής κατάστασης (steady-state error) και το σύστημα δεν θα πετύχει την επιθυμητή τιμή.

$$P = K_p \times error \quad (1)$$

**Integral (I):** Ο integral όρος λαμβάνει υπόψη του όλα τα προηγούμενα σφάλματα, εξαλείφοντας έτσι το πρόβλημα του σφάλματος σταθερής κατάστασης. Αθροίζει όλα τα προηγούμενα σφάλματα στον χρόνο που έχει περάσει και πολλαπλασιάζεται από την ολοκληρωτική ενίσχυση (integral gain) ( $K_i$ ). Ωθεί έτσι το σύστημα στο να φτάσει την επιθυμητή τιμή του προσπαθώντας να μηδενίσει το σφάλμα.

$$I = K_i \times \int error dt \quad (2)$$

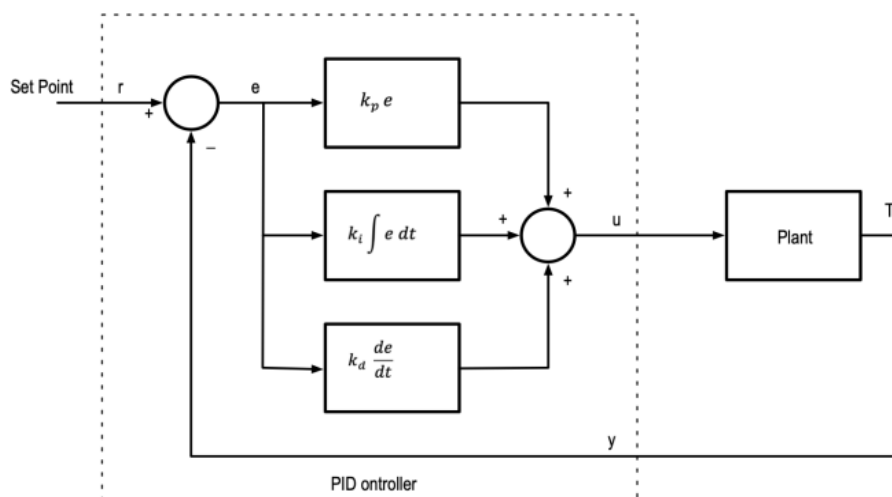
**Derivative (D):** Ο derivative όρος προβλέπει το μελλοντικό σφάλμα με βάση τον ρυθμό μεταβολής του τρέχοντος σφάλματος. Είναι μηχανισμός απόσβεσης που βοηθάει στην μείωση της υπερύψωσης (overshoot) και βελτιώνει την σταθερότητα του συστήματος με το να αντιδρά στην ταχύτητα με την οποία αλλάζει το σφάλμα. Η παραγωγική ενίσχυση (derivative gain) ( $K_d$ ) ελέγχει το αντίκτυπο του όρου αυτού. Υψηλή τιμή του  $K_d$  βοηθά στην εξομάλυνση της εξόδου του συστήματος αλλά ταυτόχρονα το κάνει επιρρεπές σε θόρυβο.

$$D = Kd \times \frac{d(\text{error})}{dt} \quad (3)$$

Ο τελικός τύπος εξόδου  $u(t)$  για τον PID ελεγκτή είναι το άθροισμα των τριών παραπάνω όρων:

$$u(t) = Kp \times \text{error} + Ki \times \int \text{error} dt + Kd \times \frac{d(\text{error})}{dt} \quad (4)$$

Για να δουλέψει σωστά ο PID εκλεκτηί πρέπει να γίνει ρύθμιση των τριών τιμών ενίσχυσης των όρων ( $Kp$ ,  $Ki$ ,  $Kd$ ) για το συγκεκριμένο σύστημα. Οι τιμές αυτές μπορούν να αλλάξουν δραματικά την συμπεριφορά του συστήματος. Με τις κατάλληλες τιμές το σύστημα μπορεί να πετύχει την επιθυμητή τιμή στον μικρότερο δυνατό χρόνο, με μηδαμινές ταλαντώσεις και καθόλου overshoot. (J. Crowe, 2005)



**Εικόνα 1.3: Μπλοκ διάγραμμα ενός PID ελεγκτή (PID Control, χ.χ.)**

Οι PID εκλεκτηί υπάρχουν σε ευρεία γκάμα συστημάτων, από καθημερινές συσκευές όπως ένας θερμοστάτης μέχρι και περίπλοκα βιομηχανικά συστήματα. Χρησιμοποιούνται στην σταθεροποίηση πτήσης τετρακόπτερων (drone) ρυθμίζοντας συνεχώς την ισχύ των ηλεκτροκινητήρων για να διατηρήσουν την ισορροπία του drone. Με βάση αυτό, το σύστημα που θα δημιουργηθεί στην παρούσα εργασία θα αποτελεί μικρό κομμάτι αυτού του ελέγχου για drone.

### 1.1.3 Ανοικτός κώδικας

Το λογισμικό ανοικτού κώδικα (open source software) είναι λογισμικό ελεύθερα διαθέσιμο σε οποιονδήποτε θέλει να το χρησιμοποιήσει ή τροποποιήσει. Σε αντίθεση με λογισμικό το οποίο ανήκει σε κάποιον και ο κώδικας του μένει κρυφός, είναι δυνατή η συνεργασία προγραμματιστών για την βελτίωση και επέκταση του λογισμικού. Είναι ευέλικτο και προσιτό θέτοντάς το ως κατάλληλη επιλογή για επιχειρήσεις ή άτομα τα οποία θέλουν να δημιουργήσουν κάτι χωρίς να μπουν στον κόπο να γράψουν κάτι από την αρχή και απλά να τροποποιήσουν κάτι έτοιμο για να ταιριάζει στην περίπτωση τους.

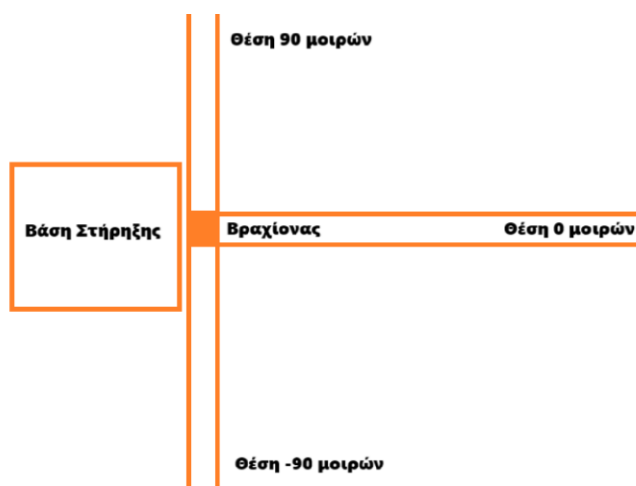


Τέτοιο λογισμικό χρησιμοποιείται σε ευρεία γκάμα εφαρμογών, από λειτουργικά συστήματα, προγράμματα για ηλεκτρονικούς υπολογιστές, εφαρμογές για κινητές συσκευές κ.α. Για παράδειγμα, ένα από τα πιο γνωστά λογισμικά ανοικτού κώδικα είναι το λειτουργικό σύστημα Linux το οποίο χρησιμοποιείται από προσωπικούς υπολογιστές μέχρι και υπερυπολογιστές, τροφοδοτώντας έτσι μεγάλο κομμάτι της υποδομής του διαδικτύου. Ένα μεγάλο πλήθος προγραμμάτων που θα χρησιμοποιηθούν στην παρούσα εργασία ανήκουν στην κατηγορία των προγραμμάτων ανοικτού κώδικα (open source program). (Perens, 2007 / 2008)

## 1.2 Το Ελικοφόρο Εκκρεμές ως εκπαιδευτικό εργαλείο για PID ελεγκτές

Το πακέτο που δημιουργήθηκε περιλαμβάνει τα αρχεία και τις οδηγίες για την κατασκευή του συστήματος, τα εκτελέσιμα αρχεία των εφαρμογών και τον κώδικα του μικροελεγκτή. Ακόμη περιλαμβάνει τον πηγαίο κώδικα των εφαρμογών για τους πιο προχωρημένους χρήστες. Ακολουθώντας τις οδηγίες, οι χρήστες θα καταφέρουν να αναδημιουργήσουν το εκπαιδευτικό εργαλείο μαζί με το περιβάλλον εφαρμογών που απαιτείται για την εκμάθηση και χρήση του PID ελεγκτή.

Το κύριο κομμάτι του εκπαιδευτικού εργαλείου είναι η ηλεκτρομηχανική διάταξη ή σύστημα που δημιουργήθηκε για την επίτευξη της εκμάθησης των PID ελεγκτών. Το σύστημα αποτελείται από ένα βραχίονα ο οποίος είναι στερεωμένος σε άρθρωση επάνω σε μία σταθερή βάση. Στην άρθρωση αυτή βρίσκεται τοποθετημένη διάταξη αισθητήρων για την μέτρηση της γωνίας που σχηματίζει ο βραχίονας με την σταθερή βάση. Στο άλλο άκρο του βραχίονα υπάρχει μηχανικά στερεωμένος ένας ηλεκτροκινητήρας που φέρει έλικα, ο οποίος παράγει ροπή κατά την λειτουργία του και αλλάζει έτσι τη γωνία του βραχίονα. Στόχος του χρήστη είναι να ελέγχει την ροπή του κινητήρα ανά πάσα στιγμή ώστε να σταθεροποιηθεί ο βραχίονας σε μια προκαθορισμένη γωνία. Αυτό γίνεται με την εφαρμογή του PID αλγορίθμου που εκτελείται τοπικά στον μικροελεγκτή που έχει επιλεγθεί. Ο μικροελεγκτής και τα ηλεκτρονικά εξαρτήματα βρίσκονται τοποθετημένα στον κλειστό χώρο της σταθερής βάσης. Περιλαμβάνουν τυπωμένη πλακέτα και συνδέσεις με καλώδια τα οποία τροφοδοτούν το σύστημα ή μεταφέρουν σήματα από τους αισθητήρες ή από το κύκλωμα οδήγησης του ηλεκτροκινητήρα στον μικροελεγκτή.



Εικόνα 1.4: Σχεδιάγραμμα του συστήματος, σημειωμένες πάνω οι πιθανές θέσεις του βραχίονα

Οι μηδέν μοίρες για τον βραχίονα θεωρήθηκαν αυθαίρετα στο σημείο που φαίνεται στην παραπάνω φωτογραφία (Εικόνα 1.4), όταν ο βραχίονας είναι παράλληλος με το επίπεδο του πατώματος, προς τα επάνω οι μοίρες αυξάνονται ενώ προς τα κάτω είναι αρνητικές. Όλο το σύστημα στερεώνεται επάνω σε γραφείο ή τραπέζι με ρυθμιζόμενη βάση στήριξης.

Ο κώδικας του μικροελεγκτή περιλαμβάνει όλες τις απαραίτητες συναρτήσεις για την επικοινωνία με το υλικό (hardware). Έχουν γραφεί και συναρτήσεις οι οποίες καθορίζουν τις λειτουργίες που μπορεί να εκτελέσει το σύστημα, οι οποίες είναι:

- Αυτόματη προστασία συστήματος: Αν για οποιαδήποτε λόγο η γωνία του βραχίονα ξεπεράσει τις 85 μοίρες προς τα θετικά, ο ηλεκτροκινητήρας παύει να λειτουργεί και διακόπτεται η λειτουργία η οποία του έδινε σήμα κίνησης. Αυτό γίνεται για λόγους ασφαλείας του χρήστη όπως και για την προστασία βλάβης του συστήματος που μπορεί να προκαλέσει στον εαυτό του.
- Χειροκίνητη λειτουργία (manual mode): σε αυτήν την λειτουργία ο χρήστης αποστέλλει τιμές για την κίνηση του ηλεκτροκινητήρα σε μορφή ποσοστού όπου το 0% σημαίνει καθόλου κίνηση και το 100% είναι η πλήρης ταχύτητα που μπορεί να αποκτήσει ο κινητήρας και άρα η κατάσταση όπου παράγει τη μέγιστη ροπή. Ο ρόλος της λειτουργίας αυτής είναι η για λόγους αποσφαλμάτωσης (debugging).
- PID λειτουργία (PID mode): Σε αυτή εκτελείται ο αλγόριθμος του PID και το σύστημα κάνει προσπάθεια σταθεροποίησης σε γωνία και τιμές  $K_p$ ,  $K_i$ ,  $K_d$  που έχουν προκαθοριστεί από τον χρήστη. Ο αλγόριθμος εκτελείται μέχρις ότου να τον σταματήσει ο χρήστης ή το σύστημα να βγει εκτός ελέγχου και να σταματήσει για λόγους ασφαλείας.
- Ζωντανή αναμετάδοση κατάστασης του συστήματος (Live System Tracking): Το σύστημα συνεχώς εκπέμπει την κατάστασή του στις συνδεδεμένες εφαρμογές σε πραγματικό χρόνο. Τα στοιχεία της κατάστασης είναι το αν εκτελείται κάποια λειτουργία ή το σύστημα είναι αδρανές, η γωνία του βραχίονα και η ταχύτητα του ηλεκτροκινητήρα.

Το Live System Tracking και η αυτόματη προστασία συστήματος εκτελούνται μονίμως ενώ οι υπόλοιπες μετά από σήμα που θα λάβει ο μικροελεγκτής από τον χρήστη.

Οι εφαρμογές που συνοδεύουν το σύστημα είναι δύο. Η μία επιτρέπει τον έλεγχο του συστήματος από κινητή συσκευή (mobile) η οποία τρέχει λογισμικό Android και επικοινωνεί με το σύστημα ασύρματα. Από την εφαρμογή αυτή μπορούν να εκτελεστούν όλες οι λειτουργίες που αναλύθηκαν παραπάνω και παρέχει τον ευκολότερο και πιο προσιτό τρόπο χρήσης του συστήματος. Η δεύτερη εφαρμογή είναι γραμμένη για ηλεκτρονικούς υπολογιστές όπου τρέχουν Windows ως λειτουργικό σύστημα. Τα χαρακτηριστικά αυτής είναι η σύνδεση του συστήματος με USB καλώδιο, γραφικό περιβάλλον το οποίο βοηθάει τον χρήστη να εκτελέσει όλες τις βασικές λειτουργίες και περιβάλλον τρισδιάστατης προβολής ενός ψηφιακού αντιγράφου του μοντέλου του συστήματος το οποίο αντικατοπτρίζει το πραγματικό. Αυτό δημιουργεί μια πιο απόμακρη προσέγγιση των λειτουργιών του συστήματος που όμως παραμένει ακράδαντα συνδεδεμένο με το πραγματικό σύστημα.

Λόγω της μορφής του πακέτου, ο κάθε χρήστης μπορεί να το τροποποιήσει σε όποιο βαθμό θέλει. Το γεγονός ότι είναι πακέτο ανοικτού κώδικα καθιστά την επεκτασιμότητα από την κοινότητα πολύ εύκολη και πιθανόν ατέλειες που έχει να είναι έναυσμα για κάποιον να τις διορθώσει για να βελτιώσει το σύστημα. Πιο έμπειροι χρήστες μπορούν να δοκιμάσουν και άλλους εκλεκτές πέρα από τον PID ή και άλλους αλγορίθμους ελέγχου όπως η ασαφής λογική, η μηχανική μάθηση κ.α.

Για την δημιουργία του πακέτου, αρχικά έγινε εκτενής έρευνα γύρω από άλλα εκπαιδευτικά εργαλεία, αφού αποφασίστηκε η μορφή του συστήματος και οι λειτουργίες που περιεγράφηκαν

παραπάνω, ακολούθησε επιλογή κατάλληλων εξαρτημάτων και προγραμμάτων για την κατασκευή και προγραμματισμό των εφαρμογών. Έχοντας κατασταλάξει σε εξαρτήματα και προγράμματα, έγινε η κατασκευή του εργαλείου και η δημιουργία του λογισμικού που το συνοδεύει. Αυτά τα βήματα θα περιγράφουν στο μήκος της εργασίας.

### 1.3 Επιλογή εξαρτημάτων

Η μεθοδολογία της επιλογής των εξαρτημάτων ακολούθησε τα εξής τρία βήματα:

1<sup>ο</sup> Βήμα: Έρευνα γύρω από το εξάρτημα, τις αρχές λειτουργίας τους, τα είδη που υπάρχουν και τις παραλλαγές τους, όπως και το εύρος κόστους.

2<sup>ο</sup> Βήμα: Ανάλυση της λειτουργίας του εξαρτήματος στο συγκεκριμένο σύστημα και τις απαιτήσεις που χρειάζονται.

3<sup>ο</sup> Βήμα: Επιλογή του κατάλληλου εξαρτήματος βάση των παραπάνω βημάτων, με κύριο στόχο το ελάχιστο κόστος και την κάλυψη των αναγκών του συστήματος σε ακρίβεια, δομή και ανθεκτικότητα.

Παρακάτω, θα γίνει παρουσίαση των επιλογών στις πιο βασικές κατηγορίες εξαρτημάτων του συστήματος ακολουθώντας τα παραπάνω βήματα.

#### 1.3.1 Μικροελεγκτές

Ένα από τα πιο βασικά εξαρτήματα που διαθέτει το σύστημα είναι ο μικροελεγκτής. Είναι υπεύθυνος για την σύνδεση των αισθητήρων και των μηχανισμών κίνησης. Σε αυτόν επίσης θα εκτελείται το λογισμικό του συστήματος και θα γίνεται και η επικοινωνία με τις εφαρμογές.

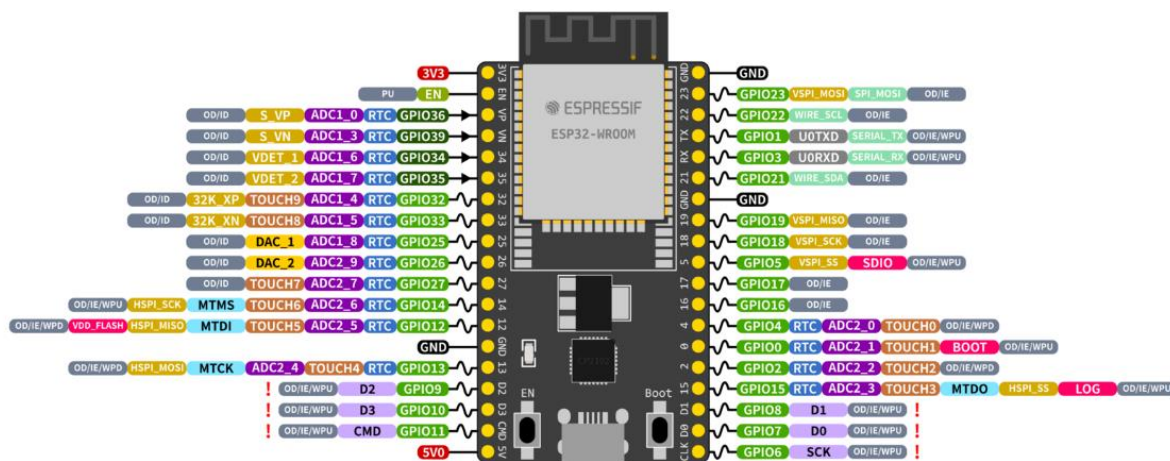
Καθώς υπάρχει τεράστια γκάμα μικροελεγκτών με πληθώρα δυνατοτήτων, πρέπει να γίνουν σαφείς οι ελάχιστες δυνατότητες που πρέπει να διαθέτει για να μπορέσει να καλύψει τις ανάγκες του συστήματος. Υπάρχει μικρή απαίτηση σε υπολογιστική ισχύ, μιας και ο PID αλγόριθμος δεν είναι τόσο σύνθετος και μπορεί να εφαρμοστεί στους περισσότερους μοντέρνους μικροελεγκτές. Ωστόσο, μιας και το σύστημα είναι σχεδιασμένο για να επεκταθεί θα ήταν καλή η επιλογή κάποιου σχετικά ισχυρού μικροελεγκτή για να μπορέσει να εφαρμόσει πιο σύνθετους αλγορίθμους ελέγχου στο μέλλον. Ακόμη απαραίτητη είναι και η ασύρματη σύνδεση μέσω Bluetooth ώστε να μπορέσει να επικοινωνεί με την mobile εφαρμογή. Παρόλο που υπάρχει γκάμα από Bluetooth module που μπορούν να δώσουν δυνατότητες Bluetooth σε οποιονδήποτε μικροελεγκτή, για την εξοικονόμηση χώρου και συνδέσεων καλό θα είναι να έχει ενσωματωμένο Bluetooth module. Οι απαιτήσεις σε αριθμό εισόδων και εξόδων γενικής χρήσεως (General Purpose Input Output: GPIO pins) είναι ελάχιστες μιας και θα χρησιμοποιηθούν μόνο δύο, μία για τον αισθητήρα και μία για το ESC.

Με βάση τους παραπάνω περιορισμούς και μετά από έρευνα, ταίριαζαν τρεις μικροελεγκτές: το ESP32, το Raspberry Pi Pico W και κάποιο από τα Nordic nRF52 Series (π.χ. nRF52840). Επιλέχθηκε το ESP32 κυρίως για το μικρό του κόστος, την κάλυψη των απαιτήσεων του συστήματος και το πλήθος παραδειγμάτων που υπάρχουν online για τον προγραμματισμό του. Το Raspberry Pi Pico W έχει παρόμοια χαρακτηριστικά με το ESP32 όμως είναι αρκετά ακριβότερο και έχει επεξεργαστή χαμηλότερης ισχύος. Το nRF52840 υπάρχει κυρίως σε αναπτυξιακές πλακέτες όπως το Arduino Nano 33 BLE το οποίο είναι αρκετά ακριβότερο από ESP32 ή σε πλακέτες όπου δεν υπάρχουν αρκετά παραδείγματα και πιθανό να είναι δύσκολος ο προγραμματισμός τους.

Το ESP32 διαθέτει έναν διπύρηνιο 32-bit Xtensa LX6 επεξεργαστή με ταχύτητες ρολογιού μέχρι και 240 MHz, 2.4 GHz Wi-Fi και Bluetooth 4.2/5.0. Υποστηρίζει πρωτόκολλα επικοινωνίας όπως το

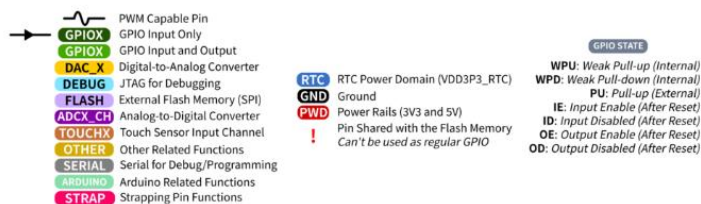
SPI, I2C, UART και δυνατότητες PWM και ADC με ανάλυση 12bit. Η αναπτυξιακή πλακέτα που επιλέχθηκε εν τέλη, είναι η ESP32 DevKitC 32D όπου έχει όλα τα χαρακτηριστικά που θα χρειαστούν στο σύστημα, έχοντας και υπόψη μελλοντικές επεκτάσεις. Ο αριθμός των GPIO pins που διαθέτει είναι πέρα από επαρκής για το συγκεκριμένο σύστημα, τροφοδοτείται με 3.3V και έχει θύρα για μεταφορά δεδομένων τύπου micro USB. (Espressif, n.d.)

ESP32-DevKitC



**ESP32 Specs**

32-bit Xtensa® dual-core @240MHz  
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz  
 Bluetooth 4.2 BR/EDR and BLE  
 520 KB SRAM (16 KB for cache)  
 448 KB ROM  
 34 GPIOs, 4x SPI, 3x UART, 2x I2C,  
 2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,  
 1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet



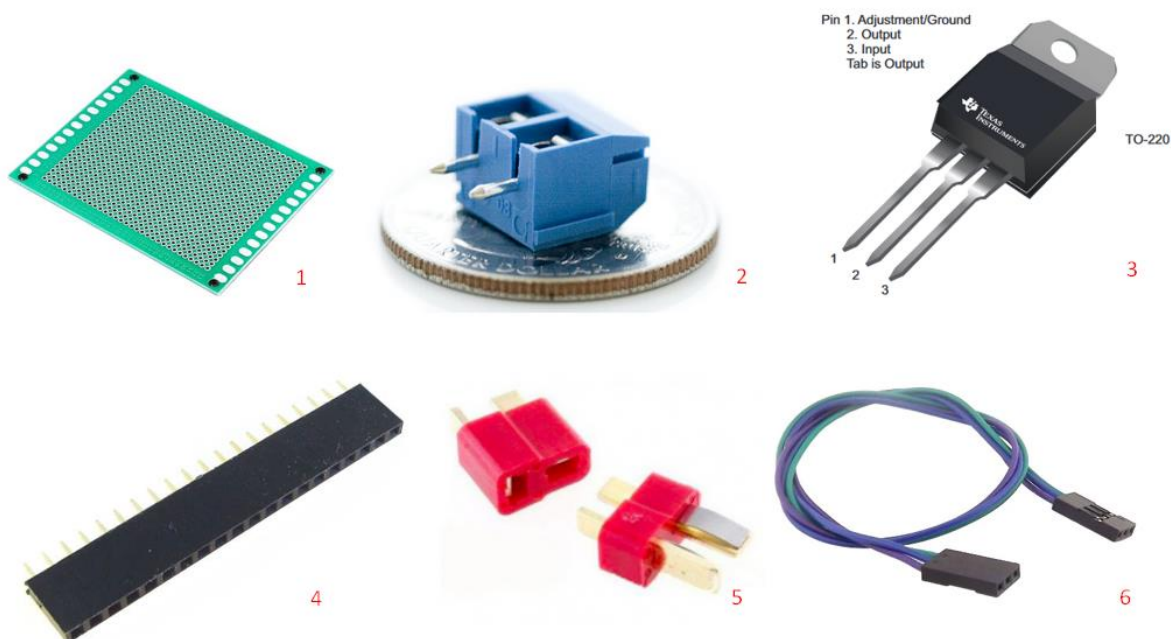
**Εικόνα 1.5: Είσοδοι και έξοδοι του ESP32 DevKitC (ESP32-DevKitC V4 , n.d.)**

**1.3.2 Ηλεκτρονικά**

Απαραίτητα για την κατασκευή του συστήματος είναι τα ηλεκτρονικά εξαρτήματα τα οποία είναι υπεύθυνα για την σύνδεση του μικροελεγκτή με τους αισθητήρες, για την τροφοδοσία και για τον έλεγχο του κινητήρα. Μερικά από τα πιο σημαντικά αναλύονται παρακάτω:

- Διάτρητη πλακέτα πρωτοτύπων (perforated prototype board): Χρησιμοποιήθηκε σε αρχικό στάδιο για την καλωδίωση όλων των υπόλοιπων εξαρτημάτων μιας και είναι εύκολη η τοποθέτηση και κόλληση ηλεκτρονικών εξαρτημάτων επάνω της. Αντικαταστάθηκε αργότερα από τυπωμένη πλακέτα.
- Τροφοδοτικό: Για την τροφοδοσία όλου του συστήματος επιλέχθηκε τροφοδοτικό γενικής χρήσης με θύρα τύπου barrel, με χαρακτηριστικά τροφοδοσίας: 12V και μέγιστο ρεύμα 5A. Η επιλογή τόσο μεγάλης τιμής ρεύματος έγινε για την εξασφάλιση της απαραίτητης ισχύος για τον ηλεκτροκινητήρα που χρησιμοποιήθηκε. Η σύνδεση με το σύστημα έγινε με αντίστοιχο barrel plug στο οποίο κολλήθηκαν καλώδια για την τροφοδοσία της πλακέτας.
- Κλέμα για πλακέτα δύο ακροδεκτών (screw terminal block): Χρησιμοποιείται για την είσοδο της τροφοδοσίας στο σύστημα και στερεώθηκε πάνω στην πλακέτα, τα καλώδια συγκρατούνται πάνω της με βοήθεια βιδών.

- Ρυθμιστής τάσης (voltage regulator) και πυκνωτές (capacitors): Για την τροφοδοσία του αισθητήρα κλίσης όπως και του ESP32 απαιτείται τάση στα 3,3V. Όπως προαναφέρθηκε, το τροφοδοτικό που χρησιμοποιήθηκε έχει έξοδο 12V. Για να μειωθεί η τάση στα 3,3V έγινε χρήση του LM 1086 voltage regulator με μέγιστο ρεύμα εξόδου τα 1,5A. Για την σωστή λειτουργία του, με βάση το datasheet του, απαιτούνται δύο πυκνωτές τόσο στην είσοδο όσο και στην έξοδό του. Χρησιμοποιήθηκαν ηλεκτρολυτικοί πυκνωτές χωρητικότητας 10μF και μέγιστης τάσης λειτουργίας 16V. Αξίζει να σημειωθεί πως κατά την λειτουργία του, το voltage regulator, ανεβάζει αρκετά την θερμοκρασία του και έτσι έγινε επένδυση σε μικρή ψήκτρα για να διανέμει την θερμότητα στον περιβάλλον χώρο με μεγαλύτερη ευκολία.
- Σειρά ακροδεκτών (Pin headers): Για την σύνδεση του μικροελεγκτή στην πλακέτα, χρησιμοποιήθηκε σειρά με pin header ώστε να είναι πιο εύκολη η αντικατάστασή του ή χρήση του σε άλλη εργασία. Αφού κολλήθηκε με καλά και κολλητήρι, η πλακέτα είχε στις σωστές αποστάσεις κατάλληλο αριθμό από θηλυκούς ακροδέκτες για να γίνει η σύνδεση με τους αρσενικούς του μικροελεγκτή. Ακόμη χρησιμοποιήθηκαν και αρσενικοί ακροδέκτες ίδιου τύπου για να γίνει η τροφοδοσία και μεταφορά σημάτων στον ελεγκτή του ηλεκτροκινητήρα και στον αισθητήρα κλίσης.
- Βύσμα τύπου T (T plug): Για την τροφοδοσία του ηλεκτροκινητήρα χρησιμοποιήθηκε ειδικό βύσμα τύπου T το οποίο χρησιμοποιείται σε πολλές παρόμοιες περιπτώσεις τροφοδοσίας drone ή άλλων τηλεκατευθυνόμενων οχημάτων μικρής κλίμακας.
- Καλώδια με ακροδέκτες: Ο αισθητήρας κλίσης συνδέεται στην πλακέτα με ειδικό καλώδιο το οποίο φέρει ακροδέκτες στα άκρα του οι οποίοι είναι συμβατοί με τα header pins που προαναφέρθηκαν.



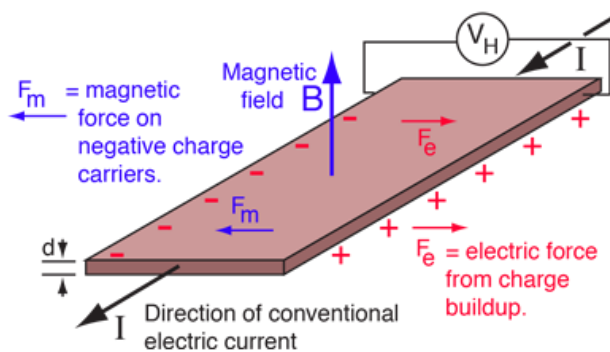
**Εικόνα 1.6: Διάτρητη πλακέτα πρωτοτύπων (1) (1528-5588-ND, n.d.), Screw terminal block (2) (Screw Terminal 2P 5mm, n.d.), Voltage Regulator (3) (LM1086 1.5-A, n.d.), Female pin header (4) (Pin Header 1x20 Female 2.54mm, n.d.), T plug (5) (T-Connector Plug, n.d.), Καλώδια με ακροδέκτη (6) (Wires 3-Pin 30cm Female to Female, n.d.)**

### 1.3.3 Αισθητήρας Κλίσης

Η ανίχνευση της γωνίας που σχηματίζει ο βραχίονας με την βάση στήριξης είναι από τις πιο βασικές υποθέσεις του συστήματος αφού είναι η μοναδική τιμή που λαμβάνει ο μικροελεγκτής ως πληροφορία για να εφαρμόσει τον PID αλγόριθμο. Καθώς πρόκειται για κινητό μέρος του συστήματος, ιδανικό θα ήταν να χρησιμοποιηθεί κάποιος τρόπος ανίχνευσης χωρίς την μηχανική σύνδεση του αισθητήρα πάνω στον βραχίονα για να διατηρηθούν οι τριβές στο ελάχιστο. Έτσι ενώ ένα απλό ποτενσιόμετρο θα επαρκούσε, έγινε έρευνα για άλλες μεθόδους ανίχνευσης της γωνίας του βραχίονα.

Εξετάστηκαν λύσεις όπως ο οπτικός κωδικοποιητής, όμως λόγω του πεπερασμένου αριθμού γωνιών που μπορεί να ανιχνεύσει δεν προσέφερε επαρκή ανάλυση ή αν κάποιος άλλος μπορούσε να πετύχει τέτοια ανάλυση ήταν μεγάλο το κόστος και έτσι αποκλείστηκε σαν επιλογή. Αισθητήρες όπου χρησιμοποιούν επιταχυνσιόμετρο ως τον τρόπο ανίχνευσης επίσης δεν επιλέχθηκαν για λόγους πολυπλοκότητας. Από τις λύσεις που υπάρχουν μόνο μία ήταν ταυτόχρονα επαρκώς αξιόπιστη, φθηνή στην αγορά και με καλή ακρίβεια, οι αισθητήρες μαγνητικού πεδίου (hall effect sensor).

Οι hall effect αισθητήρες είναι ηλεκτρονικές συσκευές όπου ανιχνεύουν την παρουσία μαγνητικών πεδίων. Έχουν τρεις ακροδέκτες, δύο για τροφοδοσία και έναν για την έξοδο σήματος. Υπάρχουν πολλά είδη hall effect αισθητήρων, για το σύστημα όμως ο πιο κατάλληλος είναι ο γραμμικός (linear) μιας και οι υπόλοιποι έχουν ως έξοδο ψηφιακό σήμα όπου δεν φέρει επαρκή πληροφορία για την ισχύ του μαγνητικού πεδίου. Οι γραμμικοί αισθητήρες, όταν υπάρχει απουσία ισχυρού μαγνητικού πεδίου έχουν ως έξοδο τάση με τιμή την μισή από αυτή της τροφοδοσίας. Όταν υπάρχει παρουσία μαγνητικού πεδίου από μαγνήτη με βόρειο ή νότιο πόλο, η τάση αυξάνεται ή μειώνεται με βάση το πόσο κοντά ή πόσο ισχυρό είναι το πεδίο. Έτσι μπορεί να γίνει μετρήσιμη όποια αλλαγή στο μαγνητικό πεδίο προκύψει εξαιτίας της απόστασης που βρίσκεται ο μαγνήτης. (Hall Effect, χ.χ.)



**Εικόνα 1.7: Σχεδιασμένο το Hall Effect, με βάση αυτό λειτουργεί ένας hall effect αισθητήρας (Hall Effect, χ.χ.)**

Για την ανίχνευση της γωνίας του βραχίονα με χρήση hall effect αισθητήρα, πρέπει να δημιουργηθεί κατάλληλη διάταξη όπου να μεταβάλλεται το μαγνητικό πεδίο κοντά στον αισθητήρα με βάση την κλίση. Για να επιτευχθεί αυτό χρησιμοποιήθηκε ένας διαμετρικός μαγνήτης, ο οποίος όταν περιστρέφεται γύρω από το κέντρο του, παρουσιάζει διαφορετικής ισχύος μαγνητικό πεδίο στην περιφέρειά του. Έτσι με συγκεκριμένη διάταξη του hall effect αισθητήρα και του μαγνήτη κολλημένο πάνω στον βραχίονα σε σημείο όπου να περιστρέφεται μαζί του, μπόρεσε και έγινε μέτρηση της γωνίας. Ο hall effect αισθητήρας που επιλέχθηκε είναι ο 49E. (LINEAR HALL-EFFECT IC, χ.χ.)



**Εικόνα 1.8: Διαμετρικός μαγνήτης που χρησιμοποιήθηκε στην διάταξη (ΜΑΓΝΗΤΗΣ ΔΙΣΚΟΣ Φ6×2,5mm ΔΙΑΜΕΤΡΙΚΟΣ, n.d.)**

#### **1.3.4 Ηλεκτροκινητήρας**

Η επιλογή του κατάλληλου ηλεκτροκινητήρα για την παραγωγή ροπής με χρήση έλικα έγινε με κριτήριο το βάρος του κινητήρα ώστε να μην επιβαρυνθεί ο βραχίονας, την ισχύ που μπορεί να δώσει και τις μέγιστες στροφές. Επιλέχθηκε έτσι brushless DC κινητήρας με χαρακτηριστικά: 1400KV ο οποίος χρησιμοποιείται σε drone, αφού καλύπτει όλες τις παραπάνω προϋποθέσεις και είναι ειδικά σχεδιασμένος για την περιστροφή έλικα.

Για να λειτουργήσει και να μπορέσει να ελεγχθεί ο κινητήρας χρειάζεται έναν ειδικό ελεγκτή ταχύτητας (electronic speed controller) (ESC). Οι brushless DC κινητήρες απαιτούν ρεύμα στις περιελίξεις τους με συγκεκριμένο τρόπο, συχνότητα και σειρά προκειμένου να λειτουργήσουν. Ο ESC παράγει τα κατάλληλα σήματα τα οποία όταν τα λάβει ο κινητήρας μπορεί να περιστραφεί με συγκεκριμένη ταχύτητα. Ακόμη ένας ESC έχει και αρκετά μέτρα προστασίας όπως από υψηλή θερμοκρασία και προστασία από τροφοδοσία με παραπάνω ρεύμα (overcurrent protection) για να μην πάθει κάποια βλάβη ο κινητήρας. Ο ESC που επιλέχθηκε ήταν από πακέτο το οποίο περιείχε τον κινητήρα, έναν έλικα, το ESC, όπως και βίδες και λοιπά εξαρτήματα για την στερέωση του έλικα και του κινητήρα. (Nagel, χ.χ.)



**Εικόνα 1.9: Ο DC brushless κινητήρας A2212 (XD A2212 1400KV Brushless Motor, χ.χ.)**

### 1.3.5 Μηχανικά μέρη

Μιας και το σύστημα αποτελείται από πολλά διαφορετικά μέρη, είναι απαραίτητος κάποιος τρόπος για να στερεωθούν μεταξύ τους ή να συνδεθούν τα κινητά μέρη έχοντας ελάχιστες τριβές. Παρακάτω τα εξαρτήματα που βοήθησαν:

- Βίδες τύπου M2, σε μήκος 6mm και 10mm: Η συγκράτηση της πλακέτας, των 3D εκτυπωμένων κομματιών και αποσπώμενων μελών του συστήματος έγινε με M2 βίδες. Επιλέχθηκαν κυρίως για την χαμηλή τιμή τους, την υψηλή διαθεσιμότητά τους και την αξιοπιστία τους. Στα περισσότερα σημεία τα 6mm ήταν αρκετά, αλλά σε μερικά ήταν απαραίτητο μεγαλύτερο μήκος σπειρώματος όπου εκεί τα 10mm ήταν κατάλληλα.
- Εμφυτεύματα με σπείρωμα (threaded inserts): Τα threaded inserts είναι μεταλλικές θήκες που έχουν σπείρωμα στον εσωτερικό τους χώρο. Τοποθετούνται σε πλαστικά μέρη όπου με χρήση θερμότητας παραμορφώνουν το πλαστικό και προσδίδουν πολύ καλή σταθερότητα στο σπείρωμα. Το σπείρωμα είναι κατάλληλο για M2 βίδες και τοποθετήθηκαν στο μεγαλύτερο μέρος των 3D εκτυπωμένων εξαρτημάτων για ένα επαναχρησιμοποιούμενο τρόπο σύνδεσής τους.
- Ρουλεμάν: Η κίνηση του βραχίονα πρέπει να γίνεται χωρίς αντιστάσεις και τριβές για να μην επηρεαστεί ο PID ελεγκτής και η διατήρηση της κατάστασης του συστήματος πρέπει να παραμείνει υγιής μετά από πολλές χρήσεις, τοποθετήθηκαν έτσι δύο ρουλεμάν δεξιά και αριστερά του βραχίονα, με διαστάσεις 22mm εξωτερική διάμετρος, 10mm εσωτερική διάμετρος και πάχος 6mm.



**Εικόνα 1.10: Το ρουλεμάν που τοποθετήθηκε στο σύστημα**

### 1.3.6 3D εκτύπωση

Το μεγαλύτερο μέρος των μηχανικών εξαρτημάτων του συστήματος είναι πρωτότυπα και φτιαγμένα ειδικά για αυτό το σκοπό. Ο καλύτερος τρόπος για την δημιουργία τέτοιων εξαρτημάτων είναι με την μέθοδο της 3D εκτύπωσης όπου αφού σχεδιαστεί το τρισδιάστατο μοντέλο του κάθε εξαρτήματος, φορτώνεται σε έναν 3D εκτυπωτή ο οποίος λιώνοντας καρούλι από πλαστικό νήμα και μετά κρύνοντας το σε κατάλληλη θέση και στιγμή μπορεί να αναδημιουργήσει το συγκεκριμένο εξάρτημα. Είναι από τις καλύτερες διαδικασίες για κατασκευή πρωτότυπων εξαρτημάτων μιας και μπορούν να γίνουν αλλαγές και τροποποιήσεις μέσα σε ώρες σε αντίθεση με άλλες μεθόδους όπου απαιτούνται μέρες. (What is 3D Printing?, χ.χ.)



Υπάρχουν τρία ευρέως χρησιμοποιούμενα είδη πλαστικών: το PLA, το ABS και το PETG. Το PLA είναι πολύ εύχρηστο όμως η μηχανική του αντοχή σε χτυπήματα δεν είναι καλή. Το ABS αντίθετα είναι πολύ σκληρό, ανθεκτικό σε χτυπήματα, αντέχει υψηλότερες θερμοκρασίες όμως είναι πολύ δύσκολη η εκτύπωσή του μιας και απαιτούνται υψηλότερες θερμοκρασίες και πολλές φορές οι 3D εκτυπωτές που μπορούν να τυπώσουν ABS είναι ακριβότεροι. Τέλος, το PETG είναι πολύ πιο εύκολο στην εκτύπωση σε σχέση με το ABS, όντας όμως πιο ελαστικό και σκληρό από το PLA είναι το κατάλληλο υλικό για την συγκεκριμένη περίπτωση. Επιλέχθηκε έτσι το PETG και σε συνδυασμό με τον εκτυπωτή της Creality CR-10, έγιναν όλες οι εκτυπώσεις των εξαρτημάτων.



**Εικόνα 1.11: Ο 3D εκτυπωτής Creality CR-10 (3D printer Creality CR-10 v2 300, n.d.)**

#### **1.4 Επιλογή προγραμμάτων**

Για την επιλογή των προγραμμάτων υπήρξαν δύο βασικά κριτήρια. Το πρώτο είναι τα προγράμματα να είναι εύχρηστα χωρίς να απαιτείται μεγάλος χρόνος εξοικείωσης με αυτά και να υπάρχει αρκετό documentation για την λειτουργία τους με παραδείγματα ή και οδηγίες χρήσης, ώστε να μπορεί ο καθένας να τροποποιήσει οποιοδήποτε τμήμα του συστήματος. Και το δεύτερο είναι να προσφέρουν δωρεάν διανομή σε όλους, ώστε να μην επιβαρυνθεί ο χρήστης με επιπλέον έξοδα. Μεγάλη προτίμηση έγινε σε προγράμματα ανοικτού κώδικα μιας και συχνά πληρούν τα δύο παραπάνω κριτήρια εξαιτίας της κοινότητας που τα υποστηρίζει. Παρακάτω θα γίνει ανάλυση όλων των προγραμμάτων που χρησιμοποιήθηκαν.

Το ESP32 μπορεί να προγραμματιστεί με πολλούς και διαφορετικούς τρόπους. Μερικοί από τους πιο γνωστούς είναι με το Arduino IDE σε C/C++, με Micropython και με το IoT Development Framework της Espressif. Η Micropython είναι μια πολύ καλή προσέγγιση για κάποιον που γνωρίζει καλή python και έχει κυρίως software γνώσεις, αφού σχετικά εύκολα μπορεί να φτιάξει πρωτότυπα. Το Arduino IDE το πιο απλό πρόγραμμα για τον προγραμματισμό του ESP32, έχει όλες τις απαραίτητες βιβλιοθήκες έτοιμες, δεν χρειάζεται μεγάλη διαδικασία για την σύνδεσή του και τέλος διαθέτει τεράστια κοινότητα που το στηρίζει με βιβλιοθήκες και παραδείγματα. Για αυτόν το λόγο επιλέχθηκε το Arduino IDE.

Προκειμένου να εκτυπωθούν τα εξαρτήματα, απαιτείται πρώτα να σχεδιαστούν σε κάποιο πρόγραμμα στον υπολογιστή, ώστε να δημιουργηθεί από αυτά κατάλληλος κώδικας ο οποίος μπορεί να εκτελεστεί από τον 3D εκτυπωτή. Μετά από έρευνα, τα πιο προσιτά προγράμματα ως προς την ευκολία στην χρήση και ως προς το κόστος τους είναι το Fusion 360 και το FreeCAD. Προφανώς υπάρχουν πολλά άλλα που είτε αναφέρονται σε επαγγελματίες και έχουν και λειτουργίες προσομοίωσης δυνάμεων, αεροδυναμικής κλπ. με αντίστοιχη όμως τιμή (π.χ. Solidworks) ή και άλλα τα οποία στοχεύουν σε χρήση από καλλιτέχνες και όχι σε μηχανικούς αφού δεν είναι κατάλληλα για μετρήσεις ακρίβειας (π.χ. Blender, Maya). Ο σχεδιασμός επιλέχθηκε να γίνει στο FreeCAD, αφού είναι ένα open source πρόγραμμα και υπερκαλύπτει τις απαιτήσεις σχεδίασης με τα εργαλεία που διαθέτει. Για την εκτύπωση, το slicer πρόγραμμα που επιλέχθηκε είναι το Ultimaker Cura.

Για τον σχεδιασμό του κυκλώματος και για την κατασκευή του PCB χρησιμοποιήθηκε το KiCAD. Ξεχώρισε ανάμεσα στα υπόλοιπα καθώς παρείχε το μεγαλύτερο αριθμό εργαλείων και παραδειγμάτων, όντας δωρεάν και open source. Προγράμματα όπως το Altium και το Eagle μπορούν να χρησιμοποιηθούν αντί αυτού, είναι όμως επί πληρωμή και στα πλαίσια της συγκεκριμένης εργασίας οι δυνατότητες του KiCAD ήταν επαρκείς.

Με την απόφαση για δημιουργία mobile εφαρμογής, αρχικά ορίστηκε το Android ως λειτουργικό σύστημα στο οποίο θα γίνει η ανάπτυξη, αφού το χρησιμοποιεί πάνω από 70% των ατόμων με κινητές συσκευές. Για την δημιουργία εφαρμογών σε Android, το Android studio προσφέρει μία ολοκληρωμένη εμπειρία με εργαλεία, Software Development Kits, προγραμματισμό σε δύο γλώσσες την Java και την Kotlin και προσφέρει επίσημη υποστήριξη από την Google. Δυνατό του χαρακτηριστικό είναι ο ενσωματωμένος προσομοιωτής Android λειτουργικού για απευθείας προβολή και εκτέλεση της εφαρμογής, όλα από το ίδιο παράθυρο.

Ενώ υπάρχουν πολλοί τρόποι για την ανάπτυξη εφαρμογών για PC και πολλές γλώσσες αντίστοιχα, όπως C# με το Microsoft Visual Studio, με Python, με C++ κ.α., η εφαρμογή χτίστηκε στο Unity που είναι σχεδιασμένο για ανάπτυξη ηλεκτρονικών παιχνιδιών (game development). Ο λόγος που επιλέχθηκε αυτή η προσέγγιση ήταν για την ευκολία που έχει το Unity στην εισαγωγή και διαχείριση 3D μοντέλων και γραφικών. Για να μπορέσει να εισαχθεί το 3D μοντέλο, ψηφιακό αντίγραφο του συστήματος, χρησιμοποιήθηκαν βιβλιοθήκες που διαθέτει το Unity. Η διεπαφή χρήστη (user interface) UI σχεδιάστηκε επίσης με το Unity χρησιμοποιώντας εργαλεία για 2D game development.

Με σκοπό την διανομή του εκπαιδευτικού πακέτου και τη χρήση του από τρίτους, καλό θα ήταν να παρουσιαστούν οι οδηγίες και τρόποι χρήσης του συστήματος για την διευκόλυνσή τους. Αυτό επιτεύχθηκε με την δημιουργία animation και την εισαγωγή του σε ένα βίντεο με οδηγίες. Τα προγράμματα που βοήθησαν σε αυτή τη διαδικασία ήταν αρχικά το Blender για το animation, ένα δωρεάν και open source πρόγραμμα με αμέτρητα εργαλεία και δυνατότητες. Για την σύνθεση του βίντεο από τα animation χρησιμοποιήθηκε πρόγραμμα επεξεργασίας βίντεο, το Davinci Resolve, όπου και αυτό είναι δωρεάν και διαθέτει όλα τα βασικά εργαλεία για απλή σύνθεση βίντεο και την εισαγωγή κειμένου οδηγιών.

## 2 ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : Κατασκευή Συστήματος

Σε αυτό το κεφάλαιο θα αναλυθούν τα βήματα που ακολουθήθηκαν για την κατασκευή του συστήματος. Αρχικά θα παρουσιαστούν πρωτότυπα σχέδια και ιδέες και ύστερα θα πάρουν μορφή. Πιο αναλυτικά, θα γίνει αναφορά στην διαδικασία της 3D σχεδίασης και εκτύπωσης, στον σχεδιασμό του ηλεκτρονικού κυκλώματος και στην πλακέτα PCB και τέλος στα βήματα κατασκευής και δοκιμών που έγιναν για τον προσδιορισμό της ορθής λειτουργίας του συστήματος με βάση τους αρχικούς στόχους. Η δομή του τελικού συστήματος θα παρουσιάζεται παράλληλα με την επεξήγηση των διαδικασιών που ακολουθήθηκαν για την κατασκευή του σε αυτό το κεφάλαιο.

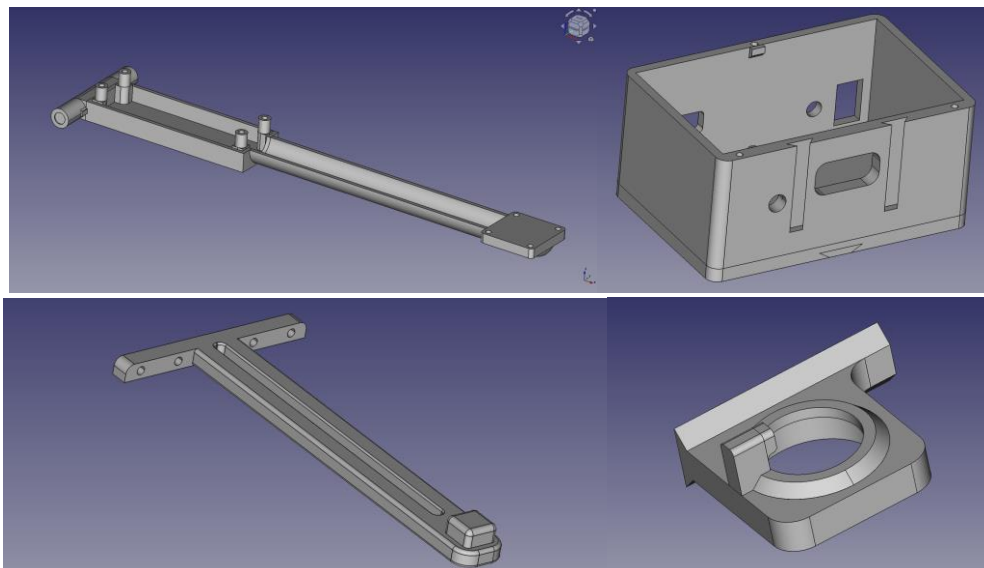
### 2.1 3D Εκτύπωση

Για την δημιουργία των περισσότερων εξαρτημάτων ακολουθήθηκε η διαδικασία της 3D εκτύπωσης. Αρχικά έγινε πρόχειρος σχεδιασμός των εξαρτημάτων σε χαρτί με πραγματική κλίμακα, έχοντας υπόψιν τα εξαρτήματα που δεν μπορούν να αλλάξουν μέγεθος μιας και είχαν επιλεγεί πρώτα, ώστε να εξασφαλιστεί ότι θα χωράνε στο τελικό σύστημα μέσα σε εσοχές. Ακολούθησε μεταφορά αυτών των σχεδίων στο πρόγραμμα του FreeCAD ώστε να δημιουργηθούν τα απαραίτητα αρχεία για την εκτύπωση. Τέλος, έγινε η εκτύπωση και τροποποιήσεις στα αρχικά σχέδια προκειμένου να διορθωθούν λάθη.

#### 2.1.1 Σχεδίαση στο FreeCAD

Για τον σχεδιασμό των κομματιών, αρχικά τοποθετήθηκαν τα ηλεκτρονικά εξαρτήματα επάνω σε χαρτί A4 και σχεδιάστηκε το κομμάτι με βάση τον χώρο που καταλάμβανε το κάθε εξάρτημα, στην περίπτωση του ηλεκτρικού μοτέρ σημειώθηκαν και τρύπες όπου θα στερεωθεί αργότερα με χρήση βιδών. Αυτό έγινε για τον προσδιορισμό κάποιων σταθερών μεγεθών στο σύστημα ώστε να προχωρήσει ο σχεδιασμός στο FreeCAD.

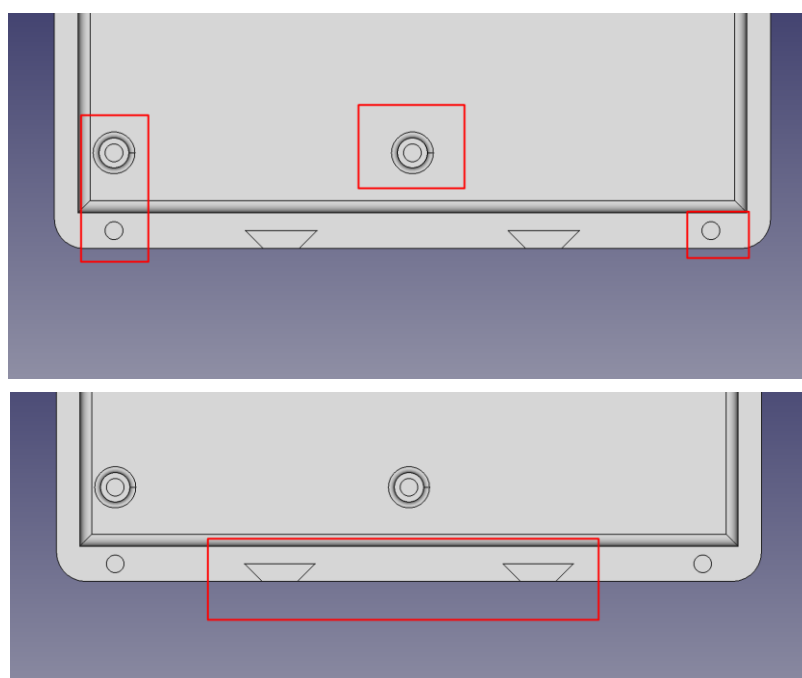
Το FreeCAD, όντας πρόγραμμα παραμετρικού σχεδιασμού, παρέχει όλα τα απαραίτητα εργαλεία για τον σχεδιασμό εξαρτημάτων όπως αυτών που απαιτούνται από το σύστημα. Για την μεταφορά των σχεδίων στο FreeCAD, αρχικά έγινε η μεταφορά των σχεδίων σε δύο διαστάσεις (2D) και αργότερα έγινε εξώθηση (extrude) των επιφανειών σε τρίτη διάσταση του χώρου ανάλογα με τα σχέδια. Για τις λεπτομέρειες του κάθε εξαρτήματος έγινε σχεδιασμός πάνω σε κάποια επιφάνεια που είχε είδη δημιουργηθεί και ακολούθησε πάλι εξώθηση στο κατάλληλο ύψος.



**Εικόνα 2.1: Μερικά από τα μοντέλα των εξαρτημάτων που σχεδιάστηκαν στο FreeCAD**

Αφού σχεδιάστηκε το κάθε εξάρτημα με τον τρόπο που προαναφέρθηκε, προστέθηκαν ομαλές γωνίες ώστε να αυξηθεί η δομική ακεραιότητα του εκάστου εξαρτήματος μετά την εκτύπωση. Ακόμη, μεγάλη προσοχή έγινε και στον προσανατολισμό του κάθε εξαρτήματος στον 3D χώρο, ώστε να εξασφαλιστεί μείωση των στηριγμάτων σε εξοχές κατά την εκτύπωση. Έτσι παρουσιάστηκε μείωση του PETG υλικού για κάθε εξάρτημα όπως και ο χρόνος εκτύπωσης.

Για την σύνδεση μερικών εξαρτημάτων μεταξύ τους, δημιουργήθηκαν ειδικές εγκοπές, και προεξοχές αντίστοιχα, στα εξαρτήματα ώστε να ταιριάζουν μεταξύ τους. Σε άλλα εξαρτήματα τα οποία θα χρειαστεί να αφαιρεί και να τοποθετεί συνεχώς ο χρήστης, σχεδιάστηκαν τρύπες για βίδες, όπως και για threaded inserts για να εισαχθούν οι βίδες. Παρακάτω ακολουθεί εικόνα περιγραφής αυτών των τρόπων σύνδεσης και στήριξης.



**Εικόνα 2.2: Παραδείγματα σύνδεσης, με βίδες και threaded inserts (επάνω), με εγκοπές (κάτω)**

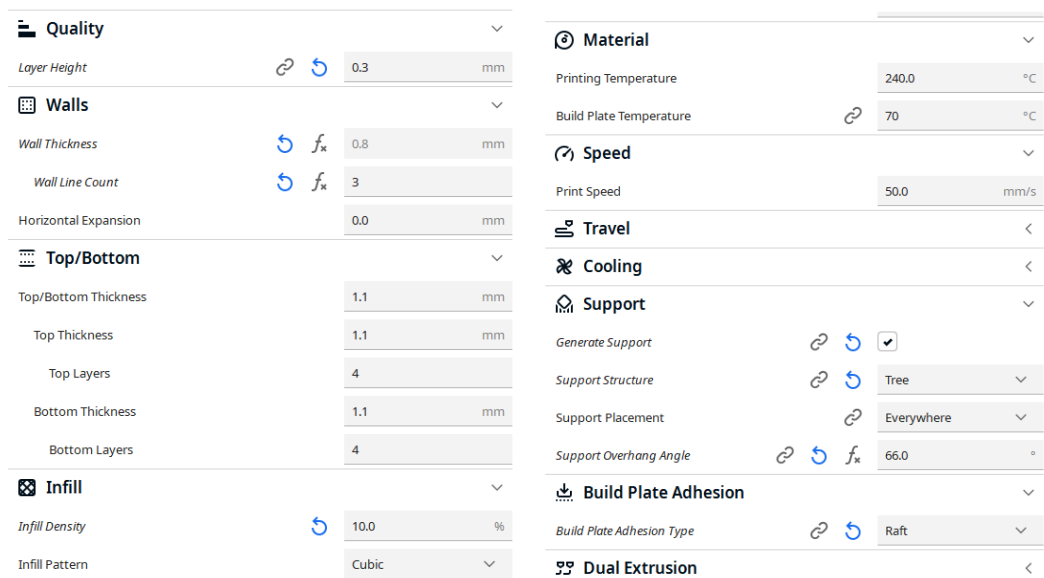
Τα εξαρτήματα που σχεδιάστηκαν είναι τα εξής:

- **main\_base:** Η κύρια βάση του συστήματος πάνω στην οποία θα στερεωθούν τα υπόλοιπα εξαρτήματα.
- **bearing\_holder** και **bearing\_holder\_plain:** πρόκειται για δύο κομμάτια τα οποία συγκρατούν τα δύο ρουλεμάν στη θέση τους και συνδέονται με το **main\_base** με τη χρήση εγκοπών. Η διαφορά των δύο είναι ότι το **\_plain** δεν έχει βάση στήριξης για τον **hall effect** αισθητήρα.
- **lid:** το καπάκι το οποίο τοποθετείται επάνω από το **main\_base** και συγκρατείται με βίδες M2 μιας και προστατεύει όλα τα ηλεκτρονικά και την πλακέτα και πρέπει να αφαιρείται εύκολα για λόγους συντήρησης και ελέγχου.
- **bottom\_shaft:** το κάτω μέρος του βραχίονα, το οποίο φέρει προεξοχές δεξιά και αριστερά για την στερέωσή του στα ρουλεμάν. Ακόμη υπάρχουν θέσεις για **threaded inserts** ώστε να βιδωθεί το πάνω μέρος του βραχίονα (**top\_shaft**) και χώρος για να τοποθετηθεί το **ESC** και τα καλώδια που συνδέονται με το **μοτέρ** και την πλακέτα.
- **top\_shaft:** το επάνω μέρος του βραχίονα το οποίο διαθέτει τρύπες για M2 βίδες για την στερέωσή του στο κάτω μέρος όπως και για να βιδωθεί επάνω του η βάση στήριξης του **μοτέρ** στο άκρο του. Υπάρχουν εγκοπές πάνω από το **ESC** για τον εξαερισμό του μιας και συμπεράθηκε πως σε συνεχή λειτουργία, αυξάνεται αρκετά η θερμοκρασία του.
- **desk\_base:** πρόκειται για κομμάτι του μηχανισμού συγκράτησης του συστήματος επάνω σε γραφείο. Βιδώνεται στο κάτω μέρος του **main\_base** με M2 βίδες και διαθέτει εγκοπή σε όλο το μήκος του ώστε να περάσει M6 βίδα και, σε συνδυασμό με άλλα εξαρτήματα, να συγκρατεί το σύστημα.
- **desk\_pad:** η επιφάνεια η οποία στερεώνεται επάνω στο **desk\_base** και με τη βοήθεια της M6 βίδας γλιστρά στον κατακόρυφο άξονα ώστε να πιέζει το γραφείο από την κάτω μεριά. Διαθέτει εσοχή για το παξιμάδι της βίδας M6.
- **screw\_head:** μικρό κομμάτι το οποίο στερεώνεται πάνω στην κεφαλή της M6 βίδας ώστε να μπορεί να βιδωθεί με το χέρι χωρίς την χρήση κάποιου εργαλείου.

Αφού ολοκληρώθηκε ο σχεδιασμός κάθε εξαρτήματος, έγινε αποθήκευση του κάθε αρχείου ξεχωριστά σε **.stl** μορφή προκειμένου να φορτωθεί στο **licer** πρόγραμμα για την παραγωγή αντίστοιχου **gcode** για τον επιλεγμένο εκτυπωτή.

### 2.1.2 Εκτύπωση

Το **licer** πρόγραμμα που επιλέχθηκε είναι το **UltiMaker Cura**, όπως αναφέρθηκε και παραπάνω. Αρχικά ενημερώθηκαν όλες οι παράμετροι στο πρόγραμμα ώστε να ταιριάξουν με τα τεχνικά χαρακτηριστικά του εκτυπωτή **Creality CR-10 v2**. Ακόμη ορίστηκαν παράμετροι για την ακρίβεια της εκτύπωσης, το πάχος των τοιχωμάτων όπως και το ποσοστό γέμισης σε εσωτερικούς χώρους. Κριτήριο για τα παραπάνω ήταν η εξασφάλιση της ανθεκτικότητας των τελικών εξαρτημάτων όπως και ταυτόχρονα η μείωση βάρους με χρήση λιγότερου υλικού σε μη κρίσιμα σημεία. Παρακάτω, μερικές από αυτές τις παραμέτρους.

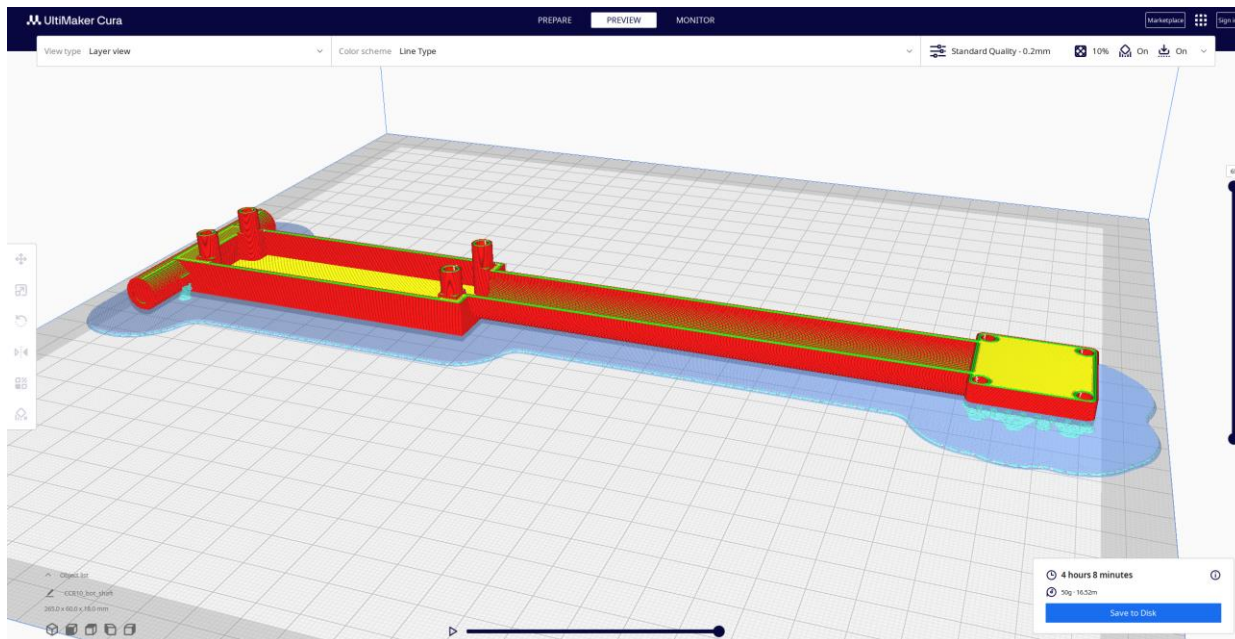


**Εικόνα 2.3: Παράμετροι που επιλέχθηκαν για τον εκτυπωτή, την εκτύπωση και το υλικό στο Cura**

Μερικές από τις επιλογές αυτές είναι:

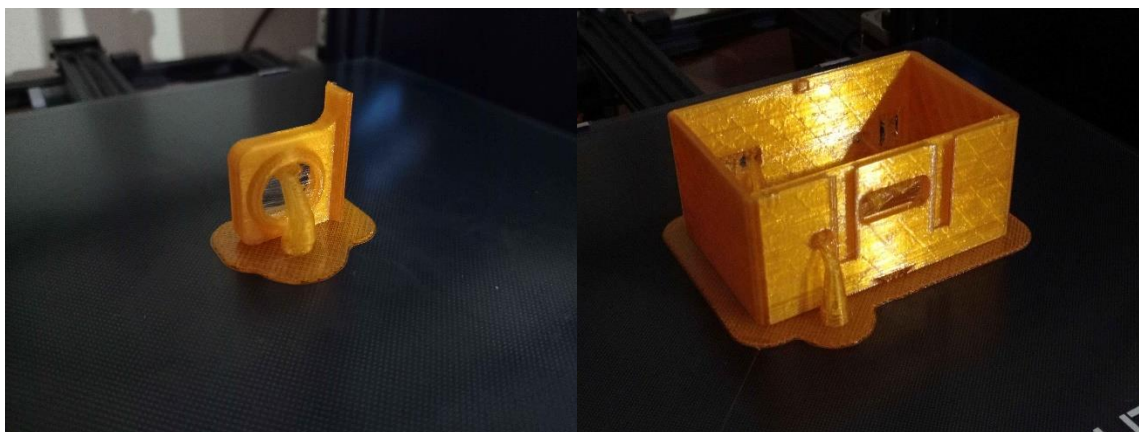
- **Wall thickness:** ρυθμίζει το πόσο παχιά θα είναι τα εξωτερικά τοιχώματα σε μεγάλα σχήματα.
- **Infill Density:** καθορίζει σε στερεά μέρη τι ποσοστό από το εσωτερικό τους θα είναι γεμάτα με υλικό, στην προκειμένη περίπτωση το 10% παρέχει ικανοποιητική δομική ακεραιότητα στο τελικό εξάρτημα με την χρήση λιγότερου υλικού και άρα χαμηλότερου βάρους.
- **Printing Temperature και Build Plate Temperature:** Η θερμοκρασία που εφαρμόστηκε για να λιώσει το PETG και να κολλήσει στο Plate το οποίο και αυτό έχει ήδη φτάσει σε συγκεκριμένη θερμοκρασία. Οι τιμές αυτές βρέθηκαν από το datasheet του PETG filament και δοκιμαστικές εκτυπώσεις ώστε να μην υπάρχουν προβλήματα κατά την τελική εκτύπωση.
- **Print Speed:** Η μέγιστη ταχύτητα με την οποία θα κινείται η κεφαλή εκτύπωσης.
- **Support Structure:** Η μορφή των υποστηρικτικών τμημάτων της εκτύπωσης ώστε να γίνει σωστή εκτύπωση σε μέρη όπου δεν στηρίζονται από την κάτω πλευρά τους.

Το επόμενο βήμα είναι η φόρτωση των .stl αρχείων, κάθε ένα ξεχωριστά, και την μετατροπή τους σε εκτελέσιμα από τον εκτυπωτή αρχεία .gcode. Το πρόγραμμα εμφανίζει και ενδεικτικό χρόνο εκτύπωσης όπως και την κατανάλωση του υλικού που θα γίνει για κάθε εξάρτημα.



**Εικόνα 2.4: Εξάρτημα στο Cura, με τον ενδεικτικό χρόνο εκτύπωσης και ποσότητα υλικού που θα χρησιμοποιηθεί κάτω δεξιά**

Τα αρχεία αυτά φορτώθηκαν σε USB Stick και ξεκίνησε η διαδικασία της εκτύπωσης. Η διαδικασία αυτή κράτησε αρκετές ημέρες, μιας και αρκετά εξαρτήματα ήταν μεγάλα σε μέγεθος. Όταν ολοκληρώθηκε η εκτύπωσή τους έγινε με μεγάλη προσοχή η αφαίρεση των στηριγμάτων (support) από τα εξαρτήματα και επιθεώρηση για τυχόν λάθη που μπορεί να προκλήθηκαν κατά την εκτύπωση.



**Εικόνα 2.5: Εξαρτήματα με τα supports τους μόλις ολοκληρώθηκε η εκτύπωση**

Με όλα τα κομμάτια έτοιμα, ακολούθησε τρίψιμο, με διαφορετικών ειδών γυαλόχαρτα, ώστε να λειανθούν οι επιφάνειες, οι οποίες κατά την σύνδεσή τους, θα εμπόδιζαν λόγω των μεγάλων ανοχών (tolerance) και τις ανακρίβειας που προέκυψε από την εκτύπωση, συνήθως συρρίκνωση.

Αξίζει να σημειωθεί ότι για την ρυθμιζόμενη βάση στήριξης του συστήματος με το γραφείο υπήρξαν δύο σχέδια, μιας και το πρώτο είχε μικρή επιτυχία. Ο τρόπος όπου στηριζόταν στο υπόλοιπο σύστημα δεν είχε αρκετή αντοχή, με αποτέλεσμα να σπάσει από την καταπόνηση που εφάρμοζε το σύστημα πάνω του. Ακόμη, ο τρόπος με τον οποίο στηριζόταν επάνω στο γραφείο ήταν με χρήση τυπωμένης βίδας, η οποία λόγω κακής σχεδίασης, δεν μπορούσε να λειτουργήσει σωστά. Για την αντιμετώπιση

αυτού του προβλήματος έγιναν αλλαγές στο σχέδιο ώστε να χρησιμοποιηθούν M2 βίδες για την στήριξη του με το σύστημα, όπως και μιας μεταφερόμενης βάσης η οποία συγκρατείται με χρήση M6 βίδας και αντίστοιχου παξιμαδιού, για την συγκράτηση στο γραφείο.



Εικόνα 2.6: Η βάση στήριξης και το ολοκληρωμένο σύστημα

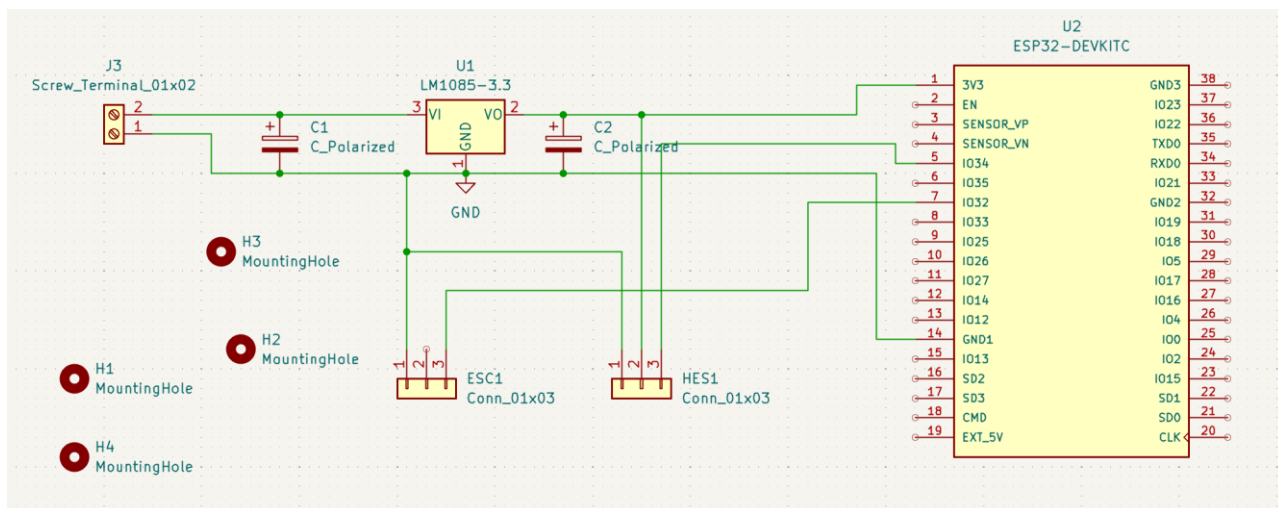
## 2.2 Πλακέτα

Σε αυτό το σημείο θα παρουσιαστεί το ηλεκτρονικό κύκλωμα, η κατασκευή του σε πρωτότυπη πλακέτα, όπως και τα βήματα για την κατασκευή τυπωμένης πλακέτας (PCB). Μετά την κατασκευή του κυκλώματος σε διάτρητη πλακέτα έγινε εμφανές ότι υπήρχαν αρκετά προβλήματα με την λειτουργία του κυκλώματος, οπότε και αποφασίστηκε να γίνει σχεδιασμός και κατασκευή ειδικής πλακέτας όπου θα εκτυπωθεί.

### 2.2.1 Ηλεκτρονικό σχέδιο

Το κύκλωμα του συστήματος έχει ως κέντρο τον μικροελεγκτή ESP32 όπου πάνω σε αυτόν έχουν συνδεθεί όλα τα υπόλοιπα ηλεκτρονικά μέρη. Αρχικά, όσον αφορά την τροφοδοσία του ESP και του ESC υπάρχει μια διαφοροποίηση στην τάση λειτουργίας. Έτσι η τροφοδοσία γίνεται στα 12V όπου καλύπτει τις ανάγκες του ESC και υπάρχει voltage regulator στα 3,3V για την τροφοδοσία του ESP και του αισθητήρα. Ο voltage regulator προτείνει την χρήση δύο πυκνωτών τόσο στην είσοδο όσο και στην έξοδό του, οι οποίοι και προστίθενται στο κύκλωμα. Για τον Hall Effect Sensor, έχει τοποθετηθεί header με τρία pins: VCC, OUT και GND όπου VCC είναι τα 3,3V τα οποία παρέχονται από το voltage regulator. Το OUT pin του hall effect συνδέεται στο GPIO34 του ESP32 το οποίο υποστηρίζει 12bit Analog to Digital Converter (ADC). Έτσι είναι δυνατή η μετατροπή της τάσης που επιστρέφει ο hall effect σε 4096 διαφορετικές τιμές ( $2^{12}$ , από 0 έως και 4095) οι οποίες παρέχουν ικανοποιητική πληροφορία για το συγκεκριμένο σύστημα. Το PWM σήμα που πρέπει να λάβει ο ESC γίνεται από το GPIO32 σε header μαζί με το GND pin ώστε να διατηρηθεί το σημείο αναφοράς ανάμεσα στα δύο κυκλώματα για την ορθή λειτουργία του ESC. Ένα barrel plug για την σύνδεση του τροφοδοτικού έχει επίσης τοποθετηθεί, όπως και ένας διακόπτης για την ενεργοποίηση και την απενεργοποίηση του συστήματος. Παρακάτω ακολουθεί εικόνα όπου απεικονίζει τις συνδέσεις αυτές.



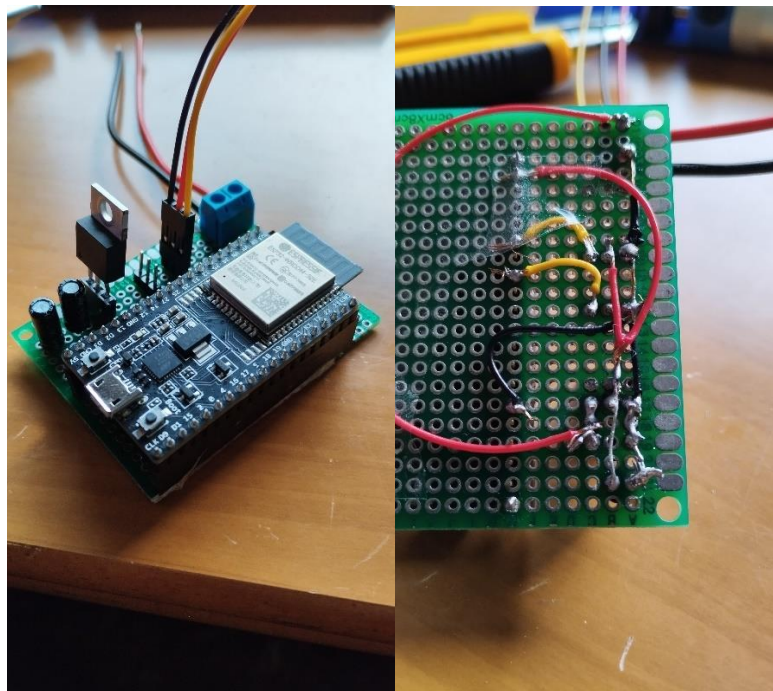


**Εικόνα 2.7: Το κύκλωμα του συστήματος, σχεδιασμένο στο KiCAD**

Μετά τον σχεδιασμό του κυκλώματος έγινε και πρόχειρη κατασκευή του σε breadboard ώστε να επιβεβαιωθεί πως η επιλογή των εξαρτημάτων, pins του ESP32 και συνδέσεις ήταν επαρκείς και σωστές για το συγκεκριμένο σύστημα. Ακολούθησαν απλές δοκιμές για διάβασμα από τον hall effect και δοκιμές με τον ESC ώστε να επαληθευτούν τα παραπάνω. Στις δοκιμές αυτές δεν αντιμετωπίστηκε κάποιο πρόβλημα.

### 2.2.2 Διάτρητη πλακέτα

Έχοντας έτοιμη την δομή του κυκλώματος, σειρά είχε η υλοποίησή του σε διάτρητη πλακέτα. Αρχικά έγινε τοποθέτηση των εξαρτημάτων επάνω σε σχηματισμό τέτοιο ώστε να διευκολυνθεί η καλωδίωση. Μετά, στερεοποιήθηκαν όλα τα εξαρτήματα στις θέσεις τους με κολλητήρι και καλά στις επαφές τους. Επόμενο βήμα ήταν να γίνουν όλες οι συνδέσεις μεταξύ τους με χρήση καλωδίων ή συνδέσεων με γραμμές από καλά. Για την στήριξη του ESP στην πλακέτα χρησιμοποιήθηκαν τα pin headers ώστε να μπορεί να αφαιρεθεί με ευκολία για χρήση σε άλλο project ή να αντικατασταθεί με άλλο σε περίπτωση βλάβης.



**Εικόνα 2.8: Το κύκλωμα σε διάτρητη πλακέτα πρωτοτύπων εμπρός και πίσω**

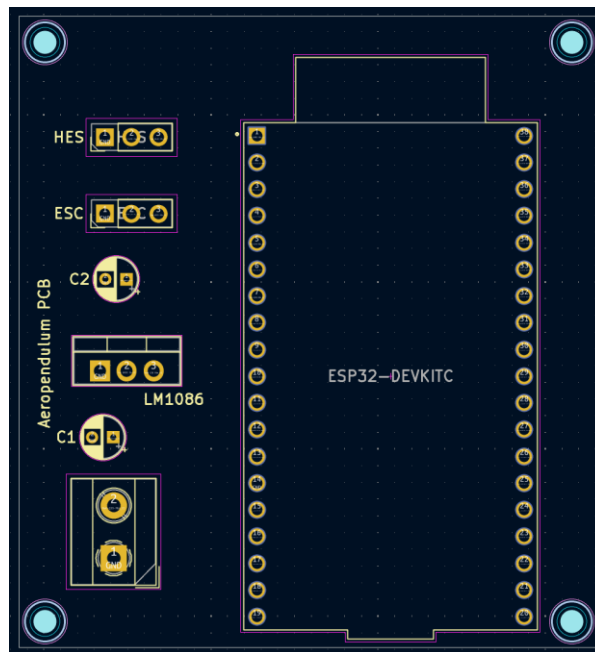
Όπως μπορεί να διακριθεί από την Εικόνα 2.8, η κάτω πλευρά της πλακέτας είναι εκτεθειμένη, πράγμα που την καθιστά μη ασφαλή και πρέπει να μετακινείται με μεγάλη προσοχή ώστε να μην δημιουργηθεί κάποιο βραχυκύκλωμα ή αποσυνδεθεί κάποιο καλώδιο. Έγινε εμφανής η ανάγκη σε μεταφορά του κυκλώματος σε κάτι καλύτερο όπως μια τυπωμένη πλακέτα η οποία θα παρέχει την απαραίτητη μόνωση στις συνδέσεις και η σταθεροποίηση των ηλεκτρονικών εξαρτημάτων θα είναι πιο στιβαρή. Επίσης θα προσδώσει και πιο επαγγελματική εμφάνιση στο τελικό σύστημα.

### 2.2.3 Σχεδίαση στο KiCAD

Το KiCAD παρέχει όλα τα απαραίτητα εργαλεία για τον σχεδιασμό και την υλοποίηση της πλακέτας. Από τον σχεδιασμό του κυκλώματος, την δημιουργία νέων σχηματικών και footprint μέχρι και τον τελικό σχεδιασμό του PCB, υπάρχουν τα κατάλληλα υποπρογράμματα. Πριν αρχίσει η διαδικασία του σχεδιασμού, δημιουργήθηκε νέο project μέσα στο KiCAD όπου και θα τοποθετηθούν τα επιμέρους αρχεία για την κατασκευή της πλακέτας. Ακολουθούν αναλυτικά τα βήματα για την σχεδίασή της.

Αρχικά επιλέχθηκε το πρόγραμμα Schematic Editor ώστε να γίνει η εισαγωγή των συμβόλων των εξαρτημάτων και να γίνουν οι συνδέσεις. Τα σχηματικά υπάρχουν στις έτοιμες βιβλιοθήκες και πολλά από αυτά έρχονται και με το αντίστοιχο footprint όπου θα χρησιμοποιηθεί αργότερα. Πέρα από τα εξαρτήματα που περιεγράφηκαν παραπάνω, προστέθηκαν και τέσσερα mounting holes τα οποία είναι απαραίτητα για την στερέωση της πλακέτας στη 3D printed βάση. Για το σχηματικό του ESP32, μιας και πρόκειται για development board, δεν υπήρχε αντίστοιχο σύμβολο μέσα στις βιβλιοθήκες του KiCAD, οπότε και έγινε η εισαγωγή του από σχηματικό που βρέθηκε online. Αφού σχεδιάστηκε το κύκλωμα, έγινε εκτέλεση του Electrical rules check (ERC) ώστε να βρεθούν πιθανά λάθη στις συνδέσεις ή τυχόν βραχυκυκλώματα. Με τη χρήση του footprint assignment tool έγινε αντιστοίχιση και έλεγχος των συμβόλων με τα αντίστοιχα footprint, προκειμένου να κατασκευαστεί η πλακέτα.

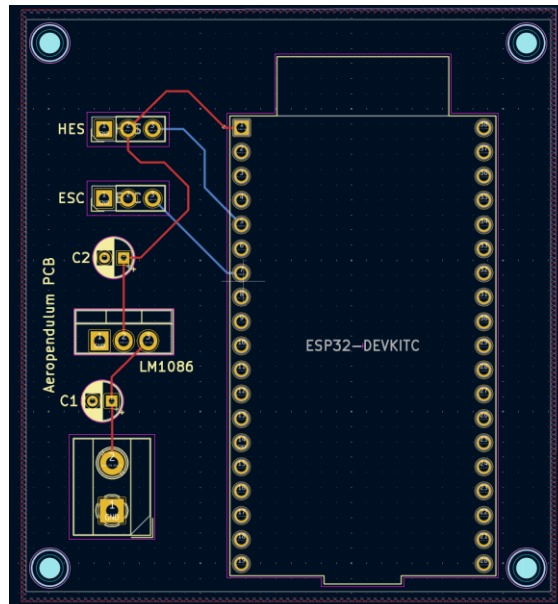
Ανοίγοντας έτσι το πρόγραμμα PCB editor, αρχικά έγινε μεταφορά των footprint από τα σχηματικά τους επιλέγοντας το Update PCB from schematic / Update PCB όπου τοποθετήθηκαν όλα τα footprints από το σχηματικό μαζεμένα. Μετά έγινε ο σχεδιασμός του περιγράμματος της πλακέτας σχεδιάζοντας ένα παραλληλόγραμμο στο EdgeCuts layer με τις σωστές διαστάσεις ώστε να μπορεί να προσαρμοστεί στα 3D printed εξαρτήματα και να περιοριστεί η επιφάνεια εργασίας. Τοποθετήθηκαν τα τέσσερα mounting holes στις ανάλογες θέσεις ώστε να ταυτίζονται με αυτές στα εκτυπωμένα εξαρτήματα. Ύστερα τοποθετήθηκαν τα υπόλοιπα εξαρτήματα σε θέσεις ώστε να γίνει εύκολα η σύνδεσή τους με traces. Ξεκινώντας από τα μεγαλύτερα σε μέγεθος εξαρτήματα και μετά στα μικρότερα ολοκληρώθηκε το layout.



Εικόνα 2.9: Το layout των εξαρτημάτων επάνω στο PCB

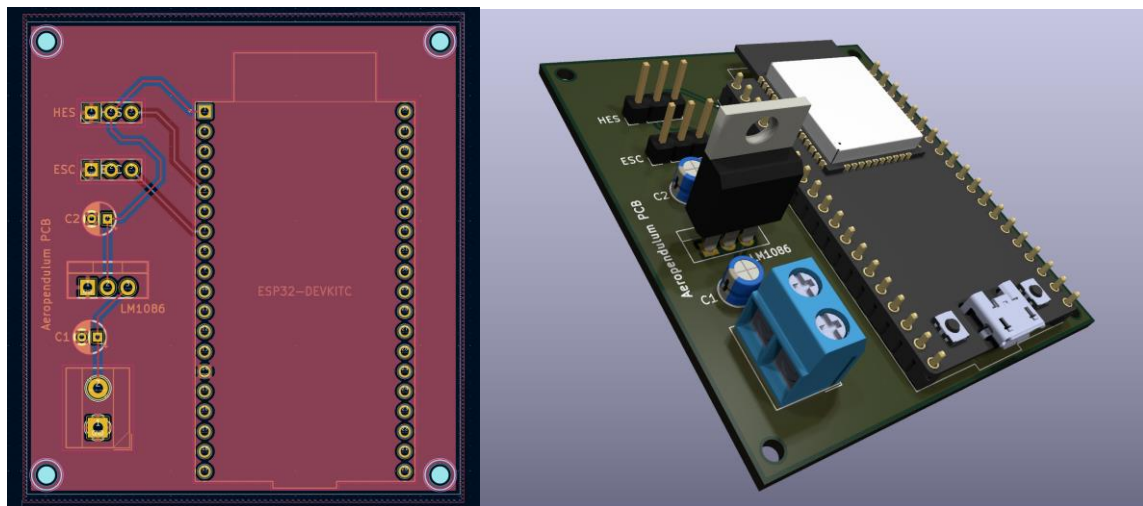
Το επόμενο βήμα είναι ο σχεδιασμός των συνδέσεων (traces). Αρχικά έπρεπε να προσδιοριστεί το πάχος που θα χρησιμοποιηθεί για τα traces. Το KiCAD προσφέρει ένα εργαλείο, το Calculator Tools, όπου μπορεί να γίνει υπολογισμός του πάχους του trace με βάση την θερμοκρασία λειτουργίας και του ρεύματος που θα το διατρέχει. Μιας και το μέγιστο ρεύμα που θα διατρέχει το trace θα είναι μικρότερο από 250mA, το οποίο είναι η μέγιστη κατανάλωση του ESP32, το calculator πρότεινε μία τιμή η οποία ήταν μικρότερη από 10mills, οπότε και επιλέχθηκαν τα 10mills μιας και είναι η default τιμή και επαρκεί για την σωστή λειτουργία της πλακέτας.

Η πλακέτα αποτελείται από δύο στρώματα χαλκού, ένα μπροστά και ένα πίσω, πάνω στα οποία μπορούν να σχεδιαστούν τα traces. Μιας και όλα τα εξαρτήματα είναι της μορφής THT (Through-Hole Technology) οι επαφές τους βρίσκονται και στις δύο πλευρές της πλακέτας, πράγμα που καθιστά τις συνδέσεις και την επιλογή layer πολύ εύκολη. Αρχικά συνδέθηκαν τα trace τροφοδοσίας και γείωσης, μιας και τα σήματα προς τον μικροελεγκτή είναι μόνο δύο. Με σωστή επιλογή των layer ανάλογα με την διαδρομή του κάθε trace, δεν χρειάστηκε να γίνει κάποια αλλαγή από layer σε layer με χρήση via αφού τα traces ήταν λίγα και η τοποθέτηση των εξαρτημάτων είχε γίνει με τέτοιο τρόπο ώστε να διευκολυνθεί η σύνδεση μεταξύ τους.



**Εικόνα 2.10:** Τα σχεδιασμένα traces επάνω στο PCB όπου ολοκληρώνουν τις συνδέσεις των εξαρτημάτων

Σαν τελευταίο βήμα πριν την εξαγωγή των αρχείων για κατασκευή, ήταν η διαδικασία του ground fill, η οποία βοηθάει στην μείωση του ηλεκτρικού θορύβου που μπορεί να διατρέξει την πλακέτα. Για να γίνει αυτό επιλέχθηκαν το καθένα layer ξεχωριστά και εφαρμόστηκε ground fill στις συνδέσεις του GND, δηλαδή όλη η επιφάνεια της πλακέτας, πέρα των υπολοίπων συνδέσεων, είναι συνδεδεμένη στο κύκλωμα του GND. Ακόμη προστέθηκε και στο εμπρός Silkscreen layer κείμενο όπου έφερε το όνομα της πλακέτας για λόγους κυρίως αισθητικούς και οργάνωσης.



**Εικόνα 2.11:** Το ολοκληρωμένο PCB (αριστερά), 3D απεικόνιση της τελικής πλακέτας με τα εξαρτήματά της (δεξιά)

Στις παραπάνω φωτογραφίες φαίνεται η τελική πλακέτα στον editor καθώς και σε φωτορεαλιστικό μοντέλο (3D render) της πραγματικής πλακέτας, το οποίο δημιουργήθηκε από εργαλείο του KiCAD ώστε να γίνει μία πρόωμη παρουσίαση της τελικής τυπωμένης πλακέτας.

## 2.2.4 Κατασκευή PCB

Με την σχεδίαση της πλακέτας στο KiCAD να έχει ολοκληρωθεί, τελευταίο βήμα είναι η εξαγωγή των Gerber αρχείων που περιγράφουν την δομή της πλακέτας προκειμένου να σταλούν σε αρμόδια υπηρεσία ώστε να γίνει η εκτύπωσή της. Επιλέχθηκε η εταιρία JLCPCB η οποία βρίσκεται στην Κίνα και έχει μεγάλη φήμη για την ποιότητα εκτύπωσης και χαμηλό κόστος. Πριν τη δημιουργία των Gerber αρχείων, έγινε εισαγωγή των δυνατοτήτων εκτύπωσης της JLCPCB, ως περιορισμοί για την πλακέτα. Οι δυνατότητες είναι τα μέγιστα και ελάχιστα μεγέθη (συνήθως διαστάσεις) τα οποία μπορεί να εφαρμόσει η εταιρία στις τεχνικές για την κατασκευή οποιασδήποτε πλακέτας και συνήθως διαφέρουν από εταιρία σε εταιρία καθώς εξαρτώνται από τις υποδομές τους. Έγινε τελικός έλεγχος με το design rule check (DRC) του KiCAD ώστε να επιβεβαιωθεί η σχεδιασμένη πλακέτα καλύπτει τις δυνατότητες της εταιρίας και θα μπορεί να τυπωθεί χωρίς κανένα πρόβλημα. Δεν υπήρξε κάποιο σφάλμα σε αυτή τη διαδικασία μιας και πρόκειται για απλό σχέδιο με λίγες συνδέσεις.

Μετά ακολουθήθηκαν βήματα της ίδιας της JLCPCB η οποία έχει αναλυτικό οδηγό για την εξαγωγή των Gerber αρχείων από το KiCAD. Αφού δημιουργήθηκαν τα αρχεία έγιναν upload στην online πλατφόρμα της εταιρίας όπου και επιλέχθηκαν περαιτέρω παράμετροι για την εκτύπωση της πλακέτας. Μερικές από αυτές είναι το πάχος της πλακέτας, το χρώμα κ.α.

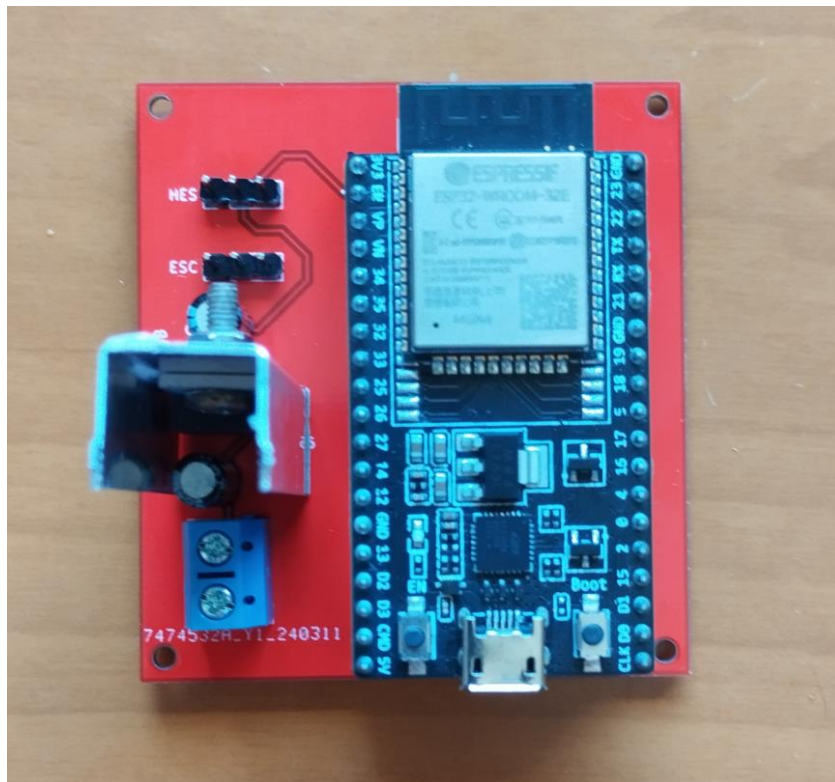
The screenshot displays the JLCPCB configuration interface with the following settings:

- Base Material:** FR-4 (selected), Flex, Aluminum, Copper Core, Rogers, PTFE Teflon.
- Layers:** 1, 2 (selected), 4, High Precision PCB, 6, 8, 10, 12, 14, 16, More.
- Dimensions:** 100 \* 100 mm.
- PCB Qty:** 5.
- Product Type:** Industrial/Consumer electronics (selected), Aerospace, Medical.
- PCB Specifications:**
  - Different Design:** 1 (selected), 2, 3, 4.
  - Delivery Format:** Single PCB (selected), Panel by Customer, Panel by JLCPCB.
  - PCB Thickness:** 0.4, 0.6, 0.8, 1.0, 1.2, 1.6 (selected), 2.0.
  - PCB Color:** Green, Purple, Red (selected), Yellow, Blue, White, Black.
  - Silkscreen:** White (selected).
  - Surface Finish:** HASL(with lead) (selected), LeadFree HASL, ENIG.
- High-spec Options:**
  - Outer Copper Weight:** 1 oz (selected), 2 oz.
  - Via Covering:** Tented (selected), Untented, Plugged, Epoxy Filled & Capped, Copper paste Filled & Capped.
  - Min via hole size/diameter:** 0.3mm(0.4/0.45mm) (selected), 0.15mm(0.25/0.3mm), 0.2mm(0.3/0.35mm), 0.25mm(0.35/0.4mm).
  - Board Outline Tolerance:** ±0.2mm(Regular) (selected), ±0.1mm(Precision).

**Εικόνα 2.12: Η φόρμα ρυθμίσεων για το PCB από την εταιρία JLCPCB**

Μιας και η πλακέτα έχει δύο layers επιλέχθηκε το '2' στα layers, το πάχος PCB στα 1,6mm, που είναι το πιο συνηθισμένο και σαν χρώμα επιλέχθηκε το κόκκινο για αισθητικούς λόγους. Οι υπόλοιπες επιλογές έμειναν στις αρχικές τους τιμές μιας και εξυπηρετούσαν τις ανάγκες της πλακέτας και κρατούσαν την τιμή χαμηλή. Ακολουθήσε παραγγελία και σε διάρκεια περίπου 2 εβδομάδων οι πλακέτες ήταν έτοιμες να μπουν στο σύστημα.

Αφού έγινε οπτικός έλεγχος των πλακετών για την εξασφάλιση σωστών συνδέσεων, κολλήθηκαν όλα τα εξαρτήματα εκ νέου επάνω στην τυπωμένη πλακέτα με την ίδια μέθοδο που χρησιμοποιήθηκε και στην διάτρητη πλακέτα.



Εικόνα 2.13: Το PCB έτοιμο με όλα τα εξαρτήματα, για να τοποθετηθεί στο σύστημα

## 2.3 Κατασκευή Συστήματος

Έχοντας τα 3D printed κομμάτια, την πλακέτα και τα υπόλοιπα εξαρτήματα έτοιμα, ακολούθησε η κατασκευή του συστήματος. Αρχικά η σύνδεση των εξαρτημάτων και μετά οι δοκιμές που έγιναν.

### 2.3.1 Κατασκευή Μηχανικού μέρους

Τα βήματα αναλυτικά για την πλήρη κατασκευή του συστήματος:

1<sup>ο</sup> Βήμα: Προσθήκη του barrel plug και του διακόπτη στο main\_base αφού έχει γίνει κόλληση καλωδίων, έτοιμα να συνδεθούν με τα υπόλοιπα ηλεκτρονικά

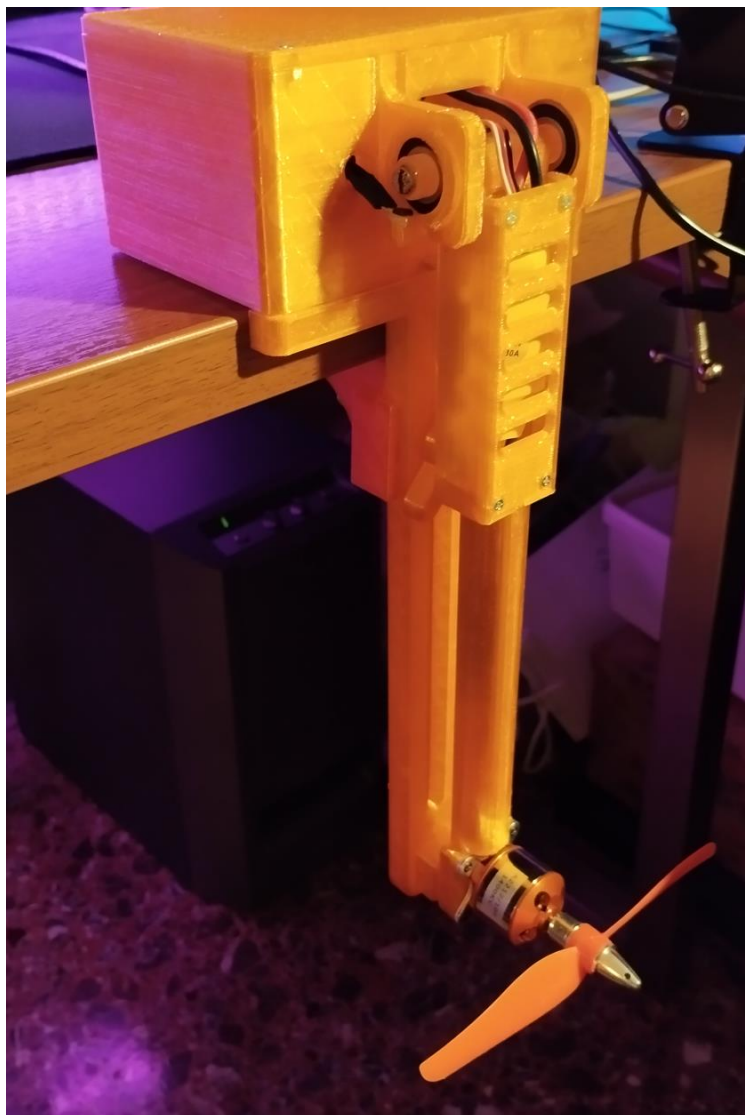
2<sup>ο</sup> Βήμα: Εισαγωγή των threaded inserts, με χρήση κολλητηριού, σε όλο το main\_base τόσο στη θέση για το PCB όσο και για το καπάκι αργότερα. Σε αυτό το βήμα προστίθεται και το PCB, στερεώνεται με βίδες και γίνονται οι απαραίτητες συνδέσεις με τα καλώδια τροφοδοσίας από τον διακόπτη.

3<sup>ο</sup> Βήμα: Εισαγωγή των ρουλεμάν στα bearing\_holder και bearing\_holder\_plain με τη χρήση θερμού αέρα για να μαλακώσει το 3D εκτυπωμένο πλαστικό και να διευκολυνθεί η τοποθέτησή τους. Χωρίς τη χρήση θερμού αέρα, όπου δοκιμάστηκε αρχικά, το πλαστικό ράγισε και έσπασε στη μέση λόγω τις πίεσης επειδή οι ανοχές ήταν μεγάλες στα εκτυπωμένα κομμάτια, τα σπασμένα αντικαταστάθηκαν αφού τυπώθηκαν ξανά.

4<sup>ο</sup> Βήμα: Σε αυτό το βήμα έγινε προετοιμασία του βραχίονα. Αρχικά στο bottom\_shaft τοποθετήθηκαν τα threaded inserts σε όλες τις τρύπες, τοποθετήθηκε το ESC και έγιναν οι ηλεκτρικές συνδέσεις με το μοτέρ. Στερεώθηκε μετά το top\_shaft από πάνω με M2 βίδες και τοποθετήθηκε το μοτέρ στην άκρη του βραχίονα με M2 βίδες οι οποίες είναι λίγο πιο μακριές από τις υπόλοιπες στα 10mm αντί για τα 6mm μιας και οι 6mm δεν επαρκούσαν να πιαστούν στα threaded inserts λόγω του πάχους των πλαστικών και της βάσης στήριξης του μοτέρ.

5<sup>ο</sup> Βήμα: Έχοντας το main\_base και τον βραχίονα έτοιμο, έγινε η σύνδεση μεταξύ τους αφού συνδέθηκαν τα ρουλεμάν μέσα στα bearing\_holders με τον βραχίονα και μετά γλίστρησαν στις εσοχές στο main\_base. Ακολούθησε η σύνδεση με το ESC και το PCB χρησιμοποιώντας το 3pin header και για την τροφοδοσία το T plug. Κολλήθηκαν καλώδια και ένα 3pin header στα τρία πόδια του hall effect αισθητήρα και ο διαμετρικός μαγνήτης στερεώθηκε στην εξοχή του βραχίονα με χρήση κόλλας. Τέλος μπήκε και το lid με M2 βίδες και ολοκληρώθηκε το σύστημα.

6<sup>ο</sup> Βήμα: Η βάση στήριξης για το γραφείο βιδώθηκε στο κάτω μέρος με M2 βίδες και threaded inserts και το desk\_pad στερεώθηκε με την M6 βίδα στο desk\_base. Στην M6 βίδα είχε από πριν στερεωθεί το screw\_head και αντίστοιχα το παξιμάδι στο desk\_pad, με κόλλα.



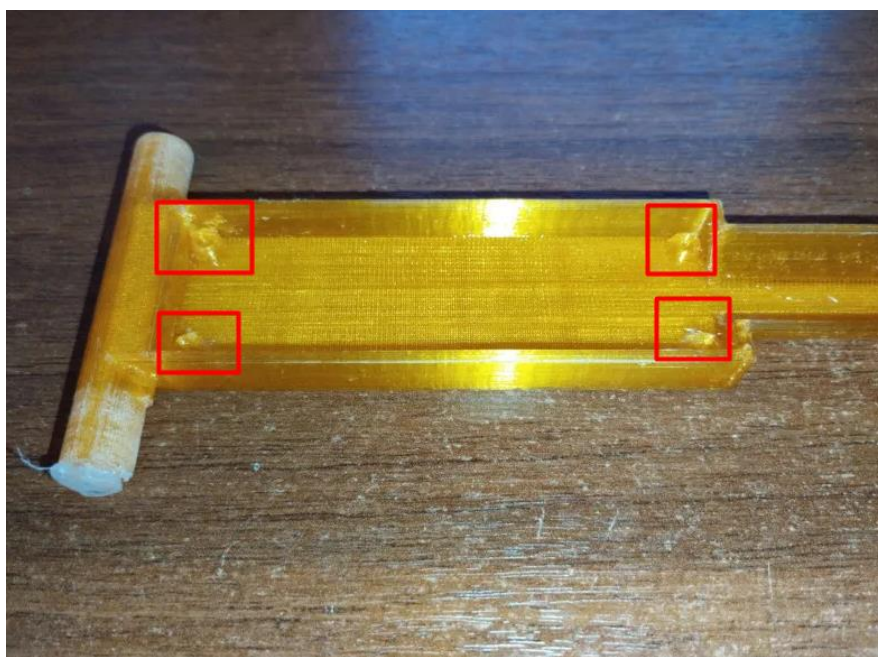
**Εικόνα 2.14: Το σύστημα ολοκληρωμένο και στερεωμένο σε γραφείο**

### 2.3.2 Έλεγχος Ηλεκτρονικών

Έχοντας το μηχανικό μέρος του συστήματος έτοιμο, πριν οποιαδήποτε συνέχεια στον προγραμματισμό του μικροελεγκτή ή των εφαρμογών, είναι απαραίτητη η δοκιμή του hardware για να διαπιστωθεί η σωστή λειτουργία του όπως και να γίνουν τυχόν αλλαγές ή τροποποιήσεις. Οι δοκιμές που ακολουθήθηκαν κάλυψαν όλες τις δυνατότητες που διαθέτει το σύστημα σε μηχανικό ηλεκτρικό και ηλεκτρονικό επίπεδο.

Αρχικά για την επιβεβαίωση της μηχανικής αντοχής του συστήματος, αφού το σύστημα στερεώθηκε σε γραφείο με τη βάση στήριξης, δοκιμάστηκε η αντοχή του στην πτώση του βραχίονα από ύψος τέτοιο όπου η γωνία του είναι η μέγιστη. Η κατάσταση αυτή είναι από τις πιο βασικές μιας και όταν το σύστημα κατά τις δοκιμές φεύγει εκτός ελέγχου θα καταλήγει πολύ συχνά σε αυτή τη θέση.

Από την θέση αυτή, αφέθηκε ο βραχίονας να πέσει με τη βοήθεια της βαρύτητας μέχρις ότου να καταλήξει στην κάτω θέση και να ακινητοποιηθεί, ύστερα έγινε έλεγχος της βάσης στήριξης και του ίδιου του βραχίονα. Όπως και είχε προαναφερθεί, σε μία από αυτές τις δοκιμές έσπασε ο βραχίονας στα σημεία όπου υπήρχαν βίδες όπως και η ίδια βάση στήριξης. Έτσι έγινε επανασχεδιασμός και των δύο, το desk mount από την αρχή εξ ολοκλήρου, μιας και δεν μπόρεσε να εκτελέσει την λειτουργία του και ο βραχίονας ενισχύθηκε στα σημεία όπου υπήρχαν θέσεις για βίδες.



Εικόνα 2.15: Τα σπασμένα μέρη μετά την πτώση, στα σημεία όπου υπήρχαν βίδες

Πριν την αρχική τροφοδοσία του συστήματος ελέγχθηκαν όλες η ηλεκτρικές συνδέσεις, με τη χρήση πολυμέτρου στην λειτουργία σύνδεσης (continuity), προκειμένου να βεβαιωθεί πως δεν υπήρξε κάποιο βραχυκύκλωμα κατά την κατασκευή και πως όλες οι συνδέσεις ήταν σωστές. Αφού τροφοδοτήθηκε για πρώτη φορά, πάλι με χρήση πολυμέτρου σε λειτουργία βολτομέτρου αυτή τη φορά, μετρήθηκαν όλες οι τάσεις τροφοδοσίας σε όλα τα πιθανά σημεία. Αρχικά μετρήθηκαν τα 12V στην είσοδο, τα 3,3V στο voltage regulator, η τάση πάνω στο ESC και η τάση στα άκρα του hall effect αισθητήρα. Ακόμη μετρήθηκε και η τάση στο OUT pin του hall effect sensor με αναφορά το GND, αφού όντας γραμμικός, η τάση χωρίς την παρουσία κάποιου μαγνητικού πεδίου θα ήταν περίπου η μισή από την τάση τροφοδοσίας (δηλαδή  $3,3V / 2 = 1,65V$ ), όπου και ήταν. Στη συνέχεια δοκιμάστηκε και η ικανότητά του στην ανίχνευση μαγνητικού πεδίου παρατηρώντας το



πως η τάση εξόδου μεταβαλλόταν όταν πλησίαζε ο βόρειος ή νότιος πόλος κάποιου μαγνήτη. Όλοι οι παραπάνω έλεγχοι είχαν τα επιθυμητά αποτελέσματα, πράγμα που υποδηλώνει πως δεν υπήρξε κάποια αποσύνδεση ή σφάλμα κατά την κατασκευή.

Όσον αφορά την σειριακή επικοινωνία, με την χρήση της Serial βιβλιοθήκης του Arduino IDE, γράφηκε απλός κώδικας ο οποίος αποστέλλει και λαμβάνει δεδομένα μέσα από την σειριακή κονσόλα του Arduino IDE. Ακόμη, με παρόμοιο κώδικα δοκιμάστηκε και η χρήση της Bluetooth επικοινωνίας κάνοντας χρήση αντίστοιχης βιβλιοθήκης του Arduino IDE για το ESP32 και ως κονσόλα χρησιμοποιήθηκε η εφαρμογή Serial Bluetooth Terminal για Android κινητά η οποία διατίθεται δωρεάν στο Play Store.

Με την σειριακή επικοινωνία να λειτουργεί, ήταν δυνατή η ανάγνωση του hall effect sensor και η προβολή των τιμών που επέστρεφε. Έτσι με κατάλληλο κώδικα, αφού ρυθμίστηκε το GPIO34 ως είσοδος στο σύστημα, έγινε εκτύπωση των τιμών του αισθητήρα στο serial console. Οι τιμές που εμφανίζονταν ήταν αυτές που παρείχε ο 12bit ADC και κυμαίνονταν από 0 έως και 4095 ανάλογα με την τάση το OUT pin του.

Τέλος δοκιμάστηκε και η λειτουργία του brushless motor και του ESC. Το ESC λειτουργεί με pulse width modulation (PWM) σήματα τα οποία θυμίζουν πολύ αυτά των servo motor. Έτσι για την διευκόλυνση του κώδικα, χρησιμοποιήθηκε η ESP32Servo βιβλιοθήκη η οποία διαθέτει όλες τις απαραίτητες εντολές για τον έλεγχο του ESC. Αφήνοντας έτσι τον βραχίονα στην κατώτατη θέση, με αύξηση του πλάτους του PWM σήματος υπήρξε και αύξηση στην ταχύτητα του έλικα και κατά επέκταση και της γωνίας του βραχίονα. Μετά από δοκιμές με διαφορετικές τιμές, συμπεράστηκε πως το μοτέρ που είχε επιλεγεί αρχικά και σε συνδυασμό με την τάση τροφοδοσίας του, ήταν πολύ ισχυρό για το συγκεκριμένο σύστημα μιας και από τις 256 διαφορετικές τιμές ταχύτητας που μπορούσε να λειτουργήσει, οποιαδήποτε τιμή πάνω από το 50 έβγαζε το σύστημα εκτός ελέγχου. Έτσι αντικαταστάθηκε με ίδιο μοτέρ χαμηλότερων KV, το αρχικό ήταν 2200KV και το νέο 1400KV. Το δεύτερο μοτέρ ήταν επίσης δυνατό για το σύστημα αλλά το εύρος των τιμών που μπορούσαν να χρησιμοποιηθούν ήταν μεγαλύτερο και ικανοποιούσαν την βασική λειτουργία του συστήματος.

Όλες οι παραπάνω δοκιμές ήταν απαραίτητες όχι μόνο για την εξακρίβωση της σωστής λειτουργίας του συστήματος, αλλά πολλές από αυτές χρησιμοποιήθηκαν και αργότερα ως βάση για τον κύριο κώδικα του μικροελεγκτή.

### 3 ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> : Κώδικας Μικροελεγκτή

Επόμενο στάδιο στην δημιουργία του συστήματος είναι η γραφή του λογισμικού του ESP32 ώστε να μπορεί λειτουργήσει όπως παρουσιάστηκε σε προηγούμενο κεφάλαιο. Αρχικά θα γίνει μία σύντομη περιγραφή για την σύνδεση του ESP32 με οποιονδήποτε υπολογιστή και πως μπορεί να προγραμματιστεί με τη χρήση του Arduino IDE. Θα γίνει ανάλυση στον κώδικα που γράφηκε για το ESP32 και θα παρουσιαστούν αλγόριθμοι και συναρτήσεις που παίζουν σημαντικό ρόλο στην λειτουργία του συστήματος.

#### 3.1 Arduino IDE

Όπως αναφέρθηκε και στο πρώτο κεφάλαιο, ο κώδικας του ESP32 θα γίνει με το Arduino IDE. Απαιτείται σαν πρώτο στάδιο όμως, να εγκατασταθούν κατάλληλοι drivers στο Windows περιβάλλον του ηλεκτρονικού υπολογιστή στον οποίο θα γίνει η ανάπτυξη, προκειμένου να μπορεί να επικοινωνήσει το ESP32 με το Arduino IDE και με οποιοδήποτε άλλο πρόγραμμα που μπορεί να χρειαστεί στη συνέχεια. Οι drivers είναι για το USB σε UART chip: CP210x και βρίσκονται στην επίσημη ιστοσελίδα τους. Αφού ακολουθηθούν τα βασικά βήματα για την εγκατάσταση του προγράμματος, υπάρχει επαρκής καθοδήγηση από το installation wizard, το driver είναι έτοιμο για χρήση. Συστήνεται και επανεκκίνηση του PC για την αποφυγή σφαλμάτων και σωστή φόρτωση του driver κατά την εκκίνηση.

Στο περιβάλλον του Arduino IDE (χρησιμοποιήθηκε η έκδοση 2.X), με την σύνδεση του ESP32 στο PC με χρήση καλωδίου USB, επιλέγεται από το μενού συνδέσεων στο πάνω μέρος του προγράμματος η αντίστοιχη COM port του ESP32 και το είδος της αναπτυξιακής πλακέτας από το αντίστοιχο πεδίο. Αν δεν εμφανιστεί η ακριβής ονομασία της πλακέτας τότε πιθανό να χρειαστεί να εγκατασταθεί η ανάλογη βιβλιοθήκη από το Board Manager του Arduino IDE. Στην παρούσα περίπτωση, έγινε επιλογή από τη λίστα μικροελεγκτών το ESP32 Dev Module. Αν ολοκληρωθούν όλα τα παραπάνω βήματα με επιτυχία, τότε το ESP32 θα είναι έτοιμο για προγραμματισμό.

Η δομή του κώδικα έχει ως εξής, υπάρχουν πολλές μεταβλητές προσβάσιμες από συναρτήσεις σε όλο το μήκος του οι οποίες αλλάζουν την κατάσταση του συστήματος ή είναι σταθερές. Έχοντας την μορφή απλού κώδικα για μικροελεγκτή στο Arduino IDE, υπάρχουν δύο πολύ βασικές συναρτήσεις, η `setup()` η οποία αρχικοποιεί την κατάσταση του συστήματος κατά την τροφοδότησή του και την συνάρτηση `loop()` όπου εκτελείται συνεχώς και καλεί άλλες συναρτήσεις που είναι υπεύθυνες για την λειτουργία του συστήματος με τον επιθυμητό τρόπο.

#### 3.2 Είσοδος / Έξοδος Δεδομένων στον Μικροελεγκτή

Το σύστημα διαθέτει ένα σήμα εισόδου, τον hall effect αισθητήρα, και ένα σήμα εξόδου, στο το ESC. Θα παρουσιαστούν με λεπτομέρειες οι τρόποι σύνδεσης των δυο παραπάνω εξαρτημάτων, τι βιβλιοθήκες χρησιμοποιήθηκαν και γιατί, όπως και το πως μπορούν να αλληλοεπιδράσουν με το υπόλοιπο πρόγραμμα.

##### 3.2.1 Είσοδος Αισθητήρα

Η ανάγνωση του αισθητήρα γίνεται με απλή ανάγνωση αναλογικής τιμής. Δηλαδή μετά την ρύθμιση για το συγκεκριμένο pin ως είσοδος, με την εντολή `analogRead()` επιτυγχάνεται φόρτωση της τρέχουσας τιμής από τον ADC. Για την απεικόνιση της τιμής σε μοίρες, απαιτείται μια μετατροπή η ΠΑΔΑ, Τμήμα Η&ΗΜ, Διπλωματική Εργασία, Κουτουλάκης Γρηγόριος

οποία γίνεται με βοήθεια συνάρτησης που γράφηκε για αυτόν το σκοπό. Η συνάρτηση φαίνεται παρακάτω και ακολουθεί εξήγηση:

```
// Reads sensor and returns angle data
void updateAngle() {
  angle_val = map((analogRead(hes_pin) - 955),0,1755,-90,90);
}
```

Κατά την ανάγνωση με την συνάρτηση `analogRead()` επιστρέφεται τιμή ανάμεσα στο 0 και στο 4095. Λόγω της απόστασης του μαγνήτη από τον hall effect αισθητήρα και των πραγματικών τιμών που μπορεί να ανιχνεύσει, το εύρος δεν θα αγγίξει ποτέ τις ακραίες αυτές τιμές (0 με 4095). Στην πραγματικότητα οι τιμές θα είναι εντός αυτού το εύρους χωρίς να αγγίζουν τα όρια. Όποιο και να είναι το εύρος όμως, πρέπει να προσαρμοστεί σε ένα νέο το οποίο να είναι από -90 έως 90 σε μοίρες, αφού αυτό είναι και το φυσικό εύρος τιμών που μπορεί να πάρει το σύστημα.

Πρόκειται για μία απλή αντιστοίχιση τιμών η οποία γίνεται με την έτοιμη συνάρτηση `map()`. Η συνάρτηση `map()` παίρνει ως τιμές: την τιμή που θα γίνει η αντιστοίχιση, την μικρότερη τιμή που παίρνει η τιμή, την μέγιστη τιμή, την μικρότερη τιμή αντιστοίχισης, την μεγαλύτερη τιμή αντιστοίχισης. Ως τιμή για αντιστοίχιση είναι η τιμή που επιστρέφεται από την `analogRead()` μείον μία σταθερή τιμή το 955. Αυτή η σταθερή τιμή είναι μία τιμή αντιστάθμισης (offset) και είναι η τιμή η οποία επιστρέφει το σύστημα όταν βρίσκεται στην θέση των -90 μοιρών. Με αυτόν τον τρόπο όταν το σύστημα βρεθεί στην θέση των -90 μοιρών θα επιστρέφεται η τιμή 0. Με το offset αυτό τώρα, όταν ο βραχίονας πάει στις 90 μοίρες επιστρέφει την τιμή 1755, άρα πλέον το εύρος 0 με 1755 μπορεί να αντιστοιχιστεί στο -90 με 90. Δηλαδή, όταν μετά το offset η τιμή είναι 0, θα επιστρέφεται -90 και όταν η τιμή είναι 1755 θα επιστρέφεται το 90, οποιαδήποτε ενδιάμεση τιμή προσαρμόζεται ανάλογα. Η τιμή της γωνίας αποθηκεύεται στην μεταβλητή `angle_val` και η συνάρτηση αυτή καλείται κάθε 25ms. Η χρονική τιμή αυτή είναι συγκεκριμένη και θα εξηγηθεί η σημασία της σε επόμενο υποκεφάλαιο.

### 3.2.2 Έξοδος στο Electronic Speed Controller

Ο τρόπος με τον οποίο γίνεται ο έλεγχος της ταχύτητας του ηλεκτροκινητήρα με το ESC είναι ίδιος με αυτόν που θα χρησιμοποιούταν και σε περίπτωση σερβοκινητήρα.

Ένας σερβοκινητήρας λειτουργεί λαμβάνοντας σήμα PWM με συγκεκριμένη συχνότητα και ανάλογα το ποσοστό του χρόνου στο οποίο υπάρχει θετικός παλμός μέσα σε έναν κύκλο της συχνότητας, αντιστοιχείται σε μία συγκεκριμένη τιμή και άρα έξοδο. Μία πολύ κοινή συχνότητα για την αποστολή παλμών σε ένα σερβοκινητήρα είναι τα 50Hz ή ένας παλμός κάθε 20ms. Ακόμη, όσον αφορά την διάρκεια όπου κάθε παλμός είναι σε ενεργή κατάσταση (όχι μηδέν), σε ένα συμβατικό σερβοκινητήρα κυμαίνεται από 1ms με 2ms. Αν, για παράδειγμα, το εύρος των γωνιών που μπορεί να πάρει είναι από 0 έως 180 μοίρες, τότε με παλμό 1ms θα δώσει έξοδο 0 μοίρες και με παλμό 2ms έξοδο 180 μοίρες. Με 1,5ms θα έχει έξοδο 90 μοίρες κ.ο.κ. Οι χρόνοι αυτοί περιγράφουν την ενεργή κατάσταση του παλμού μέσα στα 20ms.

Σε πλήρη αντιστοίχιση, ο ESC λειτουργεί με ακριβώς τα ίδια σήματα, όπου η συχνότητα είναι 50Hz ή παλμός κάθε 20ms και η διάρκεια των παλμών ελέγχου ανήκουν και αυτοί στο εύρος από 1ms με 2ms. Στην περίπτωση του ESC όμως δεν ελέγχεται η γωνία αλλά η ταχύτητα περιστροφής. Με παλμό

διάρκειας 1ms, ο κινητήρας δεν θα κινείται, ενώ με τιμή 2ms θα περιστρέφεται με την μέγιστη δυνατή ταχύτητα. Για παράδειγμα, με παλμό 1,5ms θα περιστρέφεται με ταχύτητα 50% τις μέγιστης.

Για τον προγραμματισμό αυτής της συμπεριφοράς του παλμού αρκεί μόνο η χρήση βιβλιοθήκης ελέγχου σερβοκινητήρα για το ESP32. Με την χρήση της ESP32Servo.h βιβλιοθήκης και αρχικοποίησης των τιμών σε αυτές που προαναφέρθηκαν:

```
// ESC Initialization
ESC.setPeriodHertz(50);
ESC.attach(esc_pin, 1000, 2000);
```

Όπου η πρώτη εντολή θέτει την συχνότητα και η δεύτερη το εύρος χρόνου σημάτων ελέγχου. Με κλήση της συνάρτησης write() εφαρμοσμένης πάνω στο ESC, δίνοντας τιμές από 0 έως 255, το 0 αντιστοιχεί σε παλμό 1ms και το 255 σε παλμό 2ms, μπορεί να γίνει έλεγχος της ταχύτητας του κινητήρα.

### 3.3 Πρωτόκολλο Επικοινωνίας

Καθώς το ESP32 θα βρίσκεται σε συνεχή επικοινωνία με την εφαρμογή που θα το ελέγχει, είτε αυτή είναι η mobile, είτε η εφαρμογή για PC, δεδομένα θα αποστέλλονται συνεχώς. Το ESP32 στέλνει μονίμως πληροφορίες για την κατάστασή του, ενώ λαμβάνει μόνο μετά από ενέργεια του χρήστη. Η πληροφορία αυτή επειδή είναι συγκεκριμένη, αποφασίστηκε να κωδικοποιηθεί με κάποια μέθοδο ώστε να είναι οργανωμένη και να μπορεί να χρησιμοποιηθεί και από τις δύο συσκευές που επικοινωνούν με ευκολία.

Αρχικά ερευνήθηκαν τέτοιες μέθοδοι αποστολής δεδομένων και οι δύο που ξεχώρισαν ήταν τα αρχεία JSON και απλά string χαρακτήρων όπου έχουν συγκεκριμένη δομή. Τα JSON ή JavaScript Object Notation είναι αρχεία πολύ ελαφριάς δομής όπου οργανώνουν και αποθηκεύουν την πληροφορία με ευανάγνωστο για τους ανθρώπους τρόπο (Introducing JSON, n.d.). Η μέθοδος κωδικοποίησης κειμένου μέσα σε string είναι η σε αλληλουχία (concatenation) οργάνωσή του με κάποιο διαχωριστικό σύμβολο. Από τους δύο παραπάνω τρόπους κωδικοποίησης, επιλέχθηκε η αποστολή μέσω string, για έναν πολύ βασικό λόγο, η δημιουργία JSON αρχείων παρόλο που είναι υπολογιστικά ελαφριά για έναν ηλεκτρονικό υπολογιστή, σε έναν μικροελεγκτή είναι αισθητή η διαφορά χρόνου κωδικοποίησης ενός JSON και ενός string. Λόγω της μικρής υπολογιστικής ισχύος του ESP32 (σε σχέση με ένα PC) και ταυτόχρονα την απαίτηση του να αποστέλλεται συνεχώς πληροφορία, η επιλογή των string ήταν μονόδρομος.

#### 3.3.1 Bluetooth

Η επικοινωνία του ESP32 με την mobile εφαρμογή γίνεται με την αποστολή των κωδικοποιημένων string μέσω Bluetooth. Το ESP32 διαθέτει ενσωματωμένο module Bluetooth και άρα αυτό που απομένει είναι ο προγραμματισμός του ώστε να μπορεί να εκτελέσει την παραπάνω λειτουργία. Χρησιμοποιήθηκε έτσι η βιβλιοθήκη BluetoothSerial.h στον κώδικά η οποία διαθέτει όλες τις απαραίτητες συναρτήσεις για την αποστολή και λήψη των δεδομένων.

Κατά την χρήση της, μέσα στην συνάρτηση setup() αρχικά ενεργοποιείται η λειτουργία Bluetooth με προκαθορισμένο όνομα για να μπορεί να αναγνωριστεί από άλλες συσκευές, στην συγκεκριμένη περίπτωση 'Aeropendulum'. Έτσι αν γίνει σύνδεση από εφαρμογή με το ESP32, όταν λαμβάνονται

νέα δεδομένα, γράφηκε η συνάρτηση `readData()` η οποία τα συλλέγει και τα επεξεργάζεται ανάλογα. Για την αποστολή, η συνάρτηση `sendT()` που γράφηκε στέλνει προκαθορισμένο κείμενο προς την συνδεδεμένη με αυτό εφαρμογή.

### 3.3.2 UART

Για την σύνδεση με την εφαρμογή του PC, η επικοινωνία γίνεται με σειριακή επικοινωνία μέσω UART. Πρόκειται για μία από τις πιο απλές μεθόδους επικοινωνίας μικροελεγκτών με ηλεκτρονικούς υπολογιστές αφού χρησιμοποιείται μονίμως σε παραδείγματα. Δεν χρειάζεται καμία εισαγωγή βιβλιοθήκης αφού το Arduino IDE την θεωρεί βασική. Παρόμοια με το Bluetooth, απλά γίνεται η αποστολή του κωδικοποιημένου string με την ίδια συνάρτηση `sendT()` και η λήψη γίνεται με την `readData()`. Οι δύο παραπάνω συναρτήσεις στέλνουν και διαβάζουν ταυτόχρονα την πληροφορία από όποια συσκευή και να είναι συνδεδεμένη.

### 3.3.3 Αποστολή Δεδομένων

Επόμενο βήμα είναι η εύρεση του τρόπου με τον οποίο θα γίνεται η κωδικοποίηση μέσα στο string. Αρχικά έχουμε δύο διαφορετικά πρωτόκολλα κωδικοποίησης, ένα για τα δεδομένα που αποστέλλονται από το ESP32 και ένα για αυτά που λαμβάνονται. Για την διευκόλυνση στην ανάγνωση, ως διαχωριστικά μεταξύ των μεταβλητών, δεν θα χρησιμοποιηθεί ένα συγκεκριμένο σύμβολο αλλά ένα γράμμα ανάλογα με το όνομα της μεταβλητής.

Οι μεταβλητές όπου θα αποστέλλονται είναι τρεις:

- Η γωνία του βραχίονα (γράμμα περιγραφής: A), η οποία λαμβάνει τιμές από το εύρος -90 με 90. Χρειάζονται τρεις χαρακτήρες για την αποτύπωση της τιμής και άρα οι τιμές με ένα ή δύο ψηφία θα λαμβάνουν ένα μηδενικό στην αρχή (π.χ. 17 => 017, -3 => 0-3) ώστε να διατηρηθεί το μήκος του string και άρα οι τιμές σε συγκεκριμένες θέσεις. Η προσθήκη του μηδενικού γίνεται με συνάρτηση μέσα στον κώδικα.
- Η τιμή του σήματος για τον έλεγχο της ταχύτητας του κινητήρα (γράμμα περιγραφής: P), η οποία λαμβάνει τιμές από 0 έως 100. Το 100 σε αυτή την περίπτωση δεν είναι ποσοστό, αλλά η τιμή που αποστέλλεται από το 0 έως το 255 στο ESC. Ο λόγος που δεν αποστέλλεται ποτέ το 255 ως μέγιστη τιμή έχει να κάνει με την ισχύ του συγκεκριμένου κινητήρα, θεωρήθηκε ότι το 100 από τα 255 είναι μία πολύ μεγάλη τιμή για το σύστημα και την ροπή που παράγει και άρα οι υπόλοιπες τιμές πάνω από το 100 δεν χρησιμοποιούνται καθόλου. Χρειάζονται τρεις χαρακτήρες άρα θα εφαρμοστεί πάλι η ίδια τακτική εισαγωγής μηδενικού.
- Η μεταβλητή κατάσταση (γράμμα περιγραφής: S), η οποία παίρνει τρεις τιμές: τιμή 0 σημαίνει ότι το σύστημα βρίσκεται σε αδράνεια (δεν εκτελείται κάποια λειτουργία), τιμή 1 σημαίνει ότι εκτελείται η χειροκίνητη λειτουργία και η τιμή 2 ότι εκτελείται η λειτουργία PID. Εδώ μιας και η τιμή έχει ένα ψηφίο, απαιτείται μόνο ένας χαρακτήρας και δεν αλλάζει το μήκος του, άρα εδώ δεν χρειάζεται η προσθήκη μηδενικών.

Με τις παραπάνω πληροφορίες, η τελική μορφή της κωδικοποιημένης πληροφορίας είναι η εξής:

'AxxxPxxxSx', όπου 'x' η αριθμητική τιμή της εκάστοτε μεταβλητής. Η κωδικοποίηση γίνεται με την συνάρτηση που γράφηκε για αυτόν το σκοπό η `sendData()`, η οποία είναι υπεύθυνη και για την αποστολή δεδομένων κάθε 25ms. Η αποκωδικοποίηση της πληροφορίας αυτής θα γίνεται από την συνδεδεμένη εφαρμογή, έτσι δεν υπάρχει ανάλογη συνάρτηση στο ESP32.

### 3.3.4 Λήψη Δεδομένων

Αντίθετα, η κωδικοποίηση της πληροφορίας, που λαμβάνει, γίνεται από τις εφαρμογές και άρα το ESP32 είναι υπεύθυνο για την αποκωδικοποίησή της σε έξι μεταβλητές. Αυτές οι μεταβλητές τροποποιούν το σύστημα με τέτοιο τρόπο ώστε να εκτελείται κάποια λειτουργία με συγκεκριμένες παραμέτρους. Οι μεταβλητές είναι οι εξής:

- Η επιθυμητή τιμή μεταβολής κατάστασης (γράμμα περιγραφής: S), πρόκειται για τιμή, που όπως και για την περιγραφή της κατάστασης του συστήματος, έχει τις τρεις τιμές: 0, 1 και 2. Σε αυτή την περίπτωση όμως, μιας και αυτές οι τιμές λαμβάνονται από το ESP32, ωθούν το σύστημα να αλλάξει κατάσταση ή λειτουργία. Για παράδειγμα η τιμή 0 βάζει το σύστημα σε κατάσταση αδράνειας, η τιμή 1 ενεργοποιεί την χειροκίνητη λειτουργία και η τιμή 2 εκκινεί τον αλγόριθμο του PID. Για την περίπτωση των τιμών 1 και 2, απαιτούνται και μερικά από τα παρακάτω πεδία τα οποία παίζουν τον ρόλο παραμέτρων για τις δύο αυτές λειτουργίες. Στο σύστημα όταν λαμβάνεται νέα τιμή ενημερώνεται η μεταβλητή state και έχει μήκος ενός χαρακτήρα.
- Η τιμή για την ταχύτητα κινητήρα (γράμμα περιγραφής: V), χρησιμοποιείται μόνο για την περίπτωση που εκτελείται η χειροκίνητη λειτουργία και τροποποιεί την μεταβλητή `rwm_value`. Δηλαδή όταν το σύστημα βρίσκεται σε χειροκίνητη λειτουργία, ανάλογα με την τιμή αυτή, θέτεται η τιμή σήματος ταχύτητας (παραπάνω πληροφορίες για την χειροκίνητη λειτουργία σε επόμενο κεφάλαιο). Έχει μήκος τριών χαρακτήρων.
- Τιμές για την ρύθμιση των μεταβλητών  $K_p$ ,  $K_i$  και  $K_d$  στον PID αλγόριθμο (γράμμα περιγραφής: P, I και D αντίστοιχα), τροποποιούν τις μεταβλητές που είναι υπεύθυνες για τις σταθερές αυτές. Έχουν μήκος δέκα χαρακτήρων και συμπεριλαμβάνουν προαιρετικά και το σύμβολο της τελείας (.) όπου χρησιμοποιείται ως υποδιαστολή. Χρησιμοποιούνται μόνο για την λειτουργία του PID και σε υπόλοιπες περιπτώσεις αγνοούνται.
- Η γωνία στόχος για τον PID ελεγκτή (γράμμα περιγραφής: A), πρόκειται για την μεταβλητή `target_angle` στον κώδικα. Είναι υπεύθυνη για τον ορισμό του στόχου σταθεροποίησης σε γωνίες με εύρος -90 έως 90 μοίρες (αυτό το εύρος είναι θεωρητικό μιας και οι εφαρμογές δεν επιτρέπουν την αποστολή τόσο ακραίων γωνιών για σταθεροποίηση). Έχει μήκος τριών χαρακτήρων.

Την αποκωδικοποίηση την αναλαμβάνει η συνάρτηση `decodeData()` η οποία καλείται κάθε φορά που έρχονται νέα δεδομένα. Η πλήρης μορφή ενός string για αποκωδικοποίηση είναι αυτή: `'SxVxxxPxxxxxxxxIxXXXXXXXXDXxxxxxxxxAxxx'` όπου τα 'x' αντιπροσωπεύουν τις τιμές.

## 3.4 Συναρτήσεις

Προκειμένου να οργανωθεί ο κώδικας και να μπορέσει να γίνει πιο κατανοητός από άλλους, έχουν δημιουργηθεί αρκετές συναρτήσεις οι οποίες περιέχουν συλλογή από εντολές που είναι υπεύθυνες για την εκτέλεση μιας λειτουργίας. Όπως έχει προαναφερθεί, η αποστολή δεδομένων, η κωδικοποίηση των string, η ανάγνωση και αποκωδικοποίηση της νέας πληροφορίας κ.α. έχουν οργανωθεί σε δικιές τους συναρτήσεις. Με αυτόν τον τρόπο αν αποφασιστεί να γίνει κάποια αλλαγή στον τρόπο αποστολής ή στην κωδικοποίηση, αρκεί μόνο η τροποποίηση της αντίστοιχης συνάρτησης. Οι δύο πιο βασικές συναρτήσεις στον κώδικα είναι αυτές όπου εκτελούν τις δύο βασικές λειτουργίες του: την χειροκίνητη λειτουργία και την PID λειτουργία.

### 3.4.1 Χειροκίνητη Λειτουργία

Όταν αποσταλεί κωδικοποιημένο string με την μεταβλητή κατάστασης state να έχει την τιμή 1, καλείται αυτομάτως η εντολή manual() και το σύστημα ξεκινάει την χειροκίνητη λειτουργία του. Σε αυτήν τη λειτουργία ελέγχεται συνεχώς η μεταβλητή pwm\_val και ανάλογα με αυτήν στέλνεται σήμα στον κινητήρα για να κινηθεί. Στην διάρκεια που το σύστημα βρίσκεται σε αυτήν τη λειτουργία, εξακολουθεί να αποστέλλει την κατάστασή του και να ελέγχει για νέες εντολές από την εφαρμογή. Έξοδος γίνεται μόνο αν το σύστημα βγει εκτός ελέγχου (ξεπεράσει τις 85 μοίρες ο βραχίονας) ή αν ο χρήστης μεταβάλλει την μεταβλητή state σε 0 μέσω της εφαρμογής.

### 3.4.2 Λειτουργία PID

Έχοντας εξηγήσει αναλυτικά την λειτουργία ενός PID ελεγκτή σε προηγούμενο κεφάλαιο, μένει μόνο να εφαρμοστεί και σε επίπεδο κώδικα ο οποίος θα εκτελείται από το ESP32. Όπως και με την χειροκίνητη λειτουργία, όταν ο χρήστης επιλέξει την τιμή 2 για την μεταβλητή state το σύστημα εφαρμόζει τον αλγόριθμο του PID με την εντολή PID(), διαβάζοντας πρώτα τις τιμές για τις σταθερές  $K_p$ ,  $K_i$  και  $K_d$  και την γωνία στόχου για την σταθεροποίηση. Εξακολουθεί να μεταδίδεται η κατάστασή του συστήματος κατά την εκτέλεσή του. Παρακάτω κομμάτι του κώδικα της συνάρτησης PID():

```
// PID algorithm that executes every dt
if (elapsedTime >= dt) {
  // Update Angle
  updateAngle();

  // Calculate PID
  error = target_angle - angle_val;
  integral += (error * elapsedTime);
  derivative = (angle_val - prev_angle) / elapsedTime;

  // Calculate output
  double output = Kp * error + Ki * integral + Kd * derivative;
  output = constrain(output, 0, 100);
  pwm_val = output;

  // Write output to motor
  ESC.write(pwm_val);

  // Update time variable
  prev_angle = angle_val;
}
```

Ο κώδικας αυτός μεταφράζει την λογική του PID ελεγκτή σε κώδικα. Ο κώδικας εκτελείται κάθε 25ms, δηλαδή κάθε 25ms ελέγχεται η γωνία του βραχίονα, υπολογίζονται οι τρεις όροι του PID, υπολογίζεται το άθροισμά τους και εφαρμόζεται νέο σήμα στον κινητήρα για να γίνει διόρθωση. Ο χρόνος αυτός συμπίπτει με τον χρόνο όπου αποστέλλεται πληροφορία για την κατάσταση του συστήματος. Κατά την ανάπτυξη του κώδικα, όταν αυτές οι δύο λειτουργίες δεν ήταν συγχρονισμένες, υπήρχε απρόβλεπτη συμπεριφορά του ελεγκτή καθώς υπήρχε καθυστέρηση κατά την αποστολή και το σύστημα έμενε για μερικά ms παγωμένο. Ο χρόνος κάθε επανάληψης πρέπει να

είναι αρκετά μεγάλος ώστε να προλαβαίνει να γίνεται η αποστολή των δεδομένων, αλλά όχι τόσο μεγάλος ώστε να αργεί ο ελεγκτής να κάνει διορθώσεις, αφού πρόκειται για ένα σύστημα που τρέχει σε πραγματικό χρόνο. Μετά από δοκιμές τα 25ms ήταν ικανοποιητικά. Το αποτέλεσμα του PID αλγορίθμου πρέπει να είναι ανάμεσα στις τιμές 0 και 100 ώστε να μπορεί να σταλεί το κατάλληλο σήμα στον κινητήρα, έτσι με την εντολή constrain() που χρησιμοποιήθηκε, περιορίζεται η έξοδος σε αυτές τις τιμές. Ολόκληρος ο κώδικας για το ESP32 υπάρχει στο Παράρτημα Α.



## 4 ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> : Εφαρμογή Κινητού

Σε αυτό το κεφάλαιο θα περιγραφεί η διαδικασία ανάπτυξης της mobile εφαρμογής. Αρχικά το πως λειτουργεί το Android Studio και τα εργαλεία που παρέχει και θα χρησιμοποιηθούν, μετά η διαδικασία ανάπτυξης πάνω σε κώδικα που έχει γράψει τρίτος και η λειτουργία του τρόπου σύνδεσης μέσω Bluetooth. Θα γίνει εμβάθυνση στο user interface (UI) και το πως αυτό θα προσφέρει την επιθυμητή εμπειρία στον χρήστη. Τέλος θα αναλυθούν μερικές βασικές κλάσεις οι οποίες έπαιξαν πολύ βασικό ρόλο στην τελική λειτουργία του συστήματος.

### 4.1 Android Studio

Το περιβάλλον του Android Studio προσφέρει τα απαραίτητα εργαλεία για τον προγραμματισμό εφαρμογών σε Java ή Kotlin. Με την δημιουργία ενός νέου project γίνεται η αρχική ρύθμιση της εφαρμογής, η γλώσσα που θα χρησιμοποιηθεί (Java σε αυτήν τη περίπτωση) και οι εκδόσεις του Android οι οποίες θα υποστηρίζουν την εφαρμογή. Επιλέχθηκε αρκετά παλιά έκδοση σε αυτό το σημείο ώστε να υποστηρίζεται όσο το δυνατόν μεγαλύτερος αριθμός κινητών τηλεφώνων. Αυτό ήταν εφικτό γιατί η εφαρμογή που θα αναπτυχθεί θα χρησιμοποιεί τεχνολογίες που υπήρχαν σε αυτές τις παλαιότερες εκδόσεις.

Ένα πολύ σημαντικό βήμα πριν την ανάπτυξη της εφαρμογής είναι η σύνδεση ενός φυσικού κινητού τηλεφώνου με το Android Studio ώστε να γίνουν δοκιμές και debugging σε πραγματικό χρόνο. Με χρήση ενός USB καλωδίου και μίας κινητής συσκευής η διαδικασία είναι απλή. Για να επιτευχθεί σύνδεση με το Android Studio πρέπει μονάχα να ενεργοποιηθεί από τις ρυθμίσεις του κινητού τηλεφώνου το USB Debugging. Μετά από αυτό το βήμα με κάθε αλλαγή που γίνεται στον κώδικα και επιλέγεται το Run App, εγκαθίσταται η νέα έκδοση της εφαρμογής στο κινητό, ανοίγει και είναι έτοιμη να δοκιμαστεί. Θα μπορούσε να χρησιμοποιηθεί και ο emulator που διαθέτει το Android Studio, όμως επειδή ο στόχος είναι η σύνδεση του κινητού με το ESP32 δεν μπορεί να γίνει μέσω του emulator.

Άλλο ένα πολύ χρήσιμο εργαλείο πέρα από τον κειμενογράφο (text editor) όπου γράφεται ο κώδικας, είναι το Design Window που μπορεί να ενεργοποιηθεί όταν γράφεται κώδικας σε XML για το UI. Με το Design Window είναι δυνατή η δημιουργία του UI με διαδικασία επιλογής στοιχείων από μενού με σύρσιμο και τοποθέτηση (drag and drop). Έτσι μπορεί να γίνει ολόκληρος ο σχεδιασμός του UI με επιλογή από κουμπιά, πεδία, κείμενο και να ρυθμιστεί το χρώμα τους, το μέγεθός τους, η θέση τους και το κείμενο που θα έχουν χωρίς να γραφεί καθόλου κώδικας XML.

### 4.2 Κώδικας Bluetooth

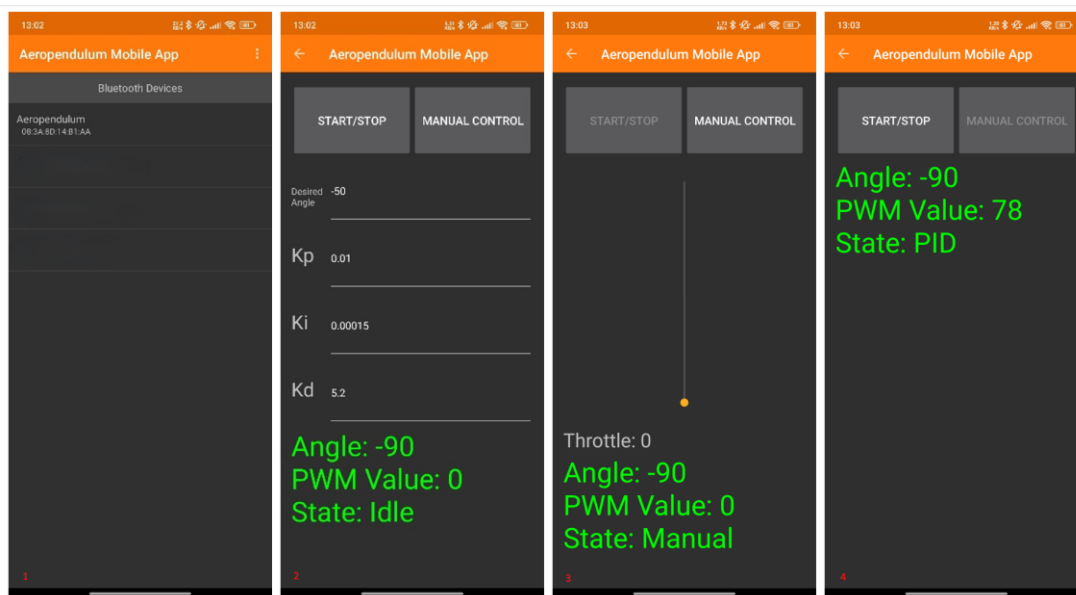
Η αρχική προσέγγιση για την δημιουργία της εφαρμογής ήταν αρχικά να γραφεί κώδικας ώστε να γίνει η επικοινωνία με το ESP32 μέσω Bluetooth σε πολύ απλό επίπεδο και μετά να επεκταθεί. Αυτό αποδείχθηκε πολύ πιο δύσκολο από όσο φαινόταν αρχικά. Η εύρεση κάποιου απλού τρόπου σύνδεσης μέσω παραδείγματος ήταν πολύ χρονοβόρα και περίπλοκη.

Η λύση σε αυτό είναι η χρήση κώδικα από τρίτους ανεβασμένος στο GitHub με ελεύθερη άδεια χρήσης. Πρόκειται για ένα project του χρήστη kai-morich ο οποίος έχει δημιουργήσει μία απλή εφαρμογή που έχει όλες τις απαραίτητες ρυθμίσεις σε επίπεδο κώδικα και συναρτήσεις για την επικοινωνία με συσκευές όπως το ESP32. Η εφαρμογή λέγεται SimpleBluetoothTerminal και

πρόκειται για κονσόλα επικοινωνίας μέσω Bluetooth Serial Port Profile (SPP). Η λογική που ακολουθήθηκε ήταν η προσαρμογή της εφαρμογής του kai-morich στο πλαίσιο των απαιτήσεων της παρούσας εφαρμογής. Διατηρήθηκαν όλα τα αρχεία όπου είναι υπεύθυνα για την επικοινωνία και την σύνδεση με Bluetooth, ενώ το UI τροποποιήθηκε αρκετά ώστε να ταιριάζει με τις απαιτήσεις που υπάρχουν από την εφαρμογή. Έτσι με αυτόν τον τρόπο εξασφαλίστηκε ένας σκελετός για τον κώδικα όπου χρησιμοποιήθηκε ως σκαλοπάτι για την ανάπτυξη μια καινούριας εφαρμογής. (kai-morich, n.d.)

### 4.3 App User Interface

Το UI της εφαρμογής είναι από τα πιο σημαντικά χαρακτηριστικά της αφού με αυτό θα αλληλοεπιδρά ο χρήστης κατά την χρήση της. Πρέπει να έχει όλες τις απαραίτητες ενδείξεις και ότι κείμενο σε κουμπί υπάρχει να είναι αντιπροσωπευτικό της λειτουργίας του. Κατά την εκκίνηση της εφαρμογής, η πρώτη όψη (view) της εφαρμογής αφορά την σύνδεση με το σύστημα. Μόλις γίνει επιτυχής σύνδεση, το επόμενο view είναι η αρχική σελίδα ρυθμίσεων και επιλογής λειτουργίας. Σε αυτό το view μπορούν να τροποποιηθούν οι παράμετροι για το PID και να ενεργοποιηθεί το PID ή η χειροκίνητη λειτουργία. Η ταχύτητα για την χειροκίνητη λειτουργία ρυθμίζεται με ένα slider μέσα από αντίστοιχο view.



**Εικόνα 4.1: Τα UI view της εφαρμογής που αναπτύχθηκε. Η επιλογή συσκευής (1), το κύριο μενού (2), το view σε χειροκίνητη λειτουργία (3), το view σε λειτουργία PID**

Για την δημιουργία των παραπάνω, χρησιμοποιήθηκε το Design Window του Android Studio κατά το μεγαλύτερο μέρος καθώς έγιναν μικρές μόνο διορθώσεις απευθείας στην XML που δημιουργήθηκε. Εκτός του πρώτου view της επιλογής συσκευής, όλα τα υπόλοιπα view πρακτικά βρίσκονται στο ίδιο αρχείο. Ανάλογα τη λειτουργία, ενεργοποιείται ή απενεργοποιείται η ορατότητα του αντίστοιχου χώρου. Ακόμη με έντονα πράσινα γράμματα είναι η ζωντανή αναμετάδοση της κατάστασης του συστήματος, με τις τρεις μεταβλητές καθαρά ορισμένες (γωνία, ταχύτητα, κατάσταση). Κατά την χειροκίνητη λειτουργία, με την χρήση του slider μπορεί να προσαρμοστεί σε πραγματικό χρόνο η ταχύτητα του κινητήρα. Όταν το σύστημα εκτελεί τον PID αλγόριθμο, δεν είναι δυνατή η τροποποίηση τιμών, όμως μπορεί να παρατηρείται η κατάσταση του.

## 4.4 Ανάπτυξη

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού και δίνει την δυνατότητα στον προγραμματιστή να οργανώνει τον κώδικά του σε κλάσεις. Οι κλάσεις είναι αρχεία στα οποία υπάρχει η πληροφορία που περιγράφει ένα αντικείμενο (object) και χρησιμοποιούνται ως βάση για την δημιουργία των object. Ο κώδικας του και-morich είναι ήδη δομημένος ανάμεσα σε αρκετές κλάσεις. Οι τροποποιήσεις έγιναν κυρίως στην κλάση TerminalFragment, όμως δημιουργήθηκαν και νέες ώστε να εξυπηρετηθούν οι ανάγκες του συστήματος.

### 4.4.1 Encode Decode Class

Μία σημαντική κλάση από αυτές που δημιουργήθηκαν είναι η StringFormatting η οποία, όπως παραπέμπει και το όνομά της, είναι υπεύθυνη για την επεξεργασία των string που λαμβάνονται και αποστέλλονται. Πιο συγκεκριμένα, αποκωδικοποιεί την κατάσταση του συστήματος που ανανεώνεται συνεχώς και την προβάλλει στην οθόνη. Ακόμη, ανάλογα με τις τιμές που έχουν τα πεδία στο κύριο view, κωδικοποιεί το κατάλληλο string για να σταλεί ώστε να εκτελεστεί η οποιαδήποτε λειτουργία με τις σωστές παραμέτρους. Καλείται από άλλες κλάσεις όπου δημιουργούν ένα αντικείμενο το οποίο έχει πρόσβαση σε ανάλογες μεθόδους για την κωδικοποίηση ή αποκωδικοποίηση.

### 4.4.2 Βασική λογική

Η βασική λογική της λειτουργίας του κύριου κώδικα (στην κλάση TerminalFragment) είναι δομημένη με τέτοιο τρόπο ώστε να ταιριάζει με τον τρόπο λειτουργίας του κώδικα του ESP32 και με πρόθεση να μπορεί να λειτουργήσει σε συνεργασία με την εφαρμογή στο PC χωρίς να επηρεάζει η μία την άλλη.

Αρχικά, ρυθμίζονται όλα τα πεδία, τα κουμπιά και γενικά όλα τα στοιχεία με το κατάλληλο κείμενό τους και συνδέονται σε μεταβλητές. Οι μεταβλητές αυτές αλλάζουν ανάλογα με την τιμή των πεδίων, και ανά πάσα στιγμή έχουν την σωστή παράμετρο για τον PID αλγόριθμο. Ακολουθούν εντολές που περιγράφουν την λειτουργία και την συμπεριφορά των κουμπιών. Με το πάτημα τους, αποστέλλεται κωδικοποιημένο το string στο ESP32 το οποίο δρα αναλόγως.

Ιδιαίτερο ενδιαφέρον έχει η εναλλαγή των view ανάλογα με την λειτουργία που εκτελείται. Το ίδιο το σύστημα επηρεάζει την κατάσταση της εφαρμογής ανάλογα με την κατάστασή του. Δηλαδή όταν επιλέγεται ένα κουμπί το οποίο βάζει το σύστημα να εκτελέσει κάποια λειτουργία, δεν αλλάζει το view από το ίδιο το κουμπί. Το σύστημα αφού λάβει την εντολή, θα την αποκωδικοποιήσει και θα ξεκινήσει να εκτελεί την λειτουργία του, αλλάζοντας πρώτα την μεταβλητή state η οποία καθορίζει την κατάσταση του. Η εφαρμογή εφόσον λαμβάνει συνεχώς την κατάσταση του συστήματος, θα παρατηρήσει την αλλαγή αυτή και θα ενεργοποιήσει το κατάλληλο view. Με αυτόν τον τρόπο εξασφαλίζεται ότι όποια εφαρμογή από τις δύο στείλει κάποιο σήμα, θα αντιδράσει με τον σωστό τρόπο.

### 4.4.3 Χτίσιμο Εφαρμογής

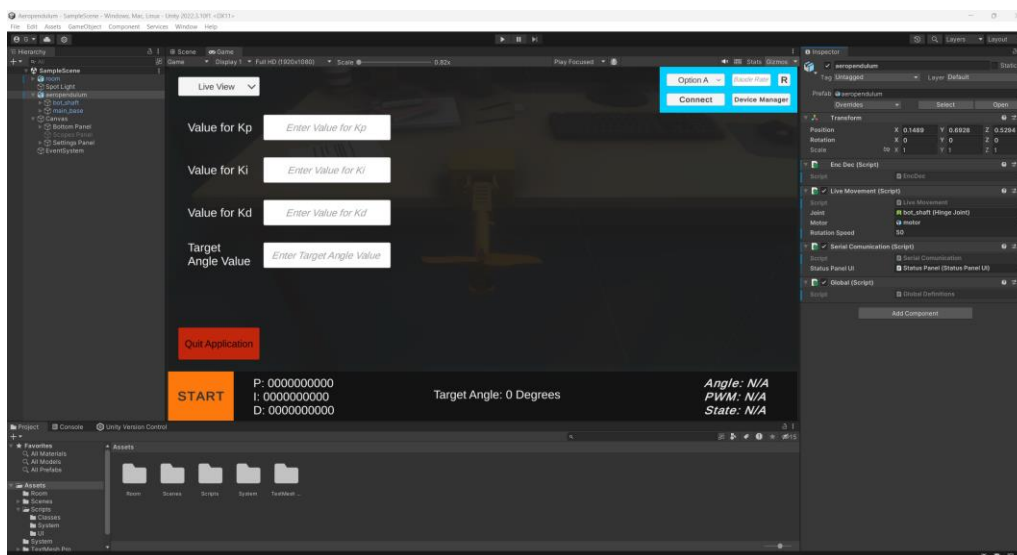
Κατά την ολοκλήρωση της ανάπτυξης της εφαρμογής, ακολουθεί η διαδικασία του build κατά την οποία δημιουργείται ένα αρχείο μορφής .apk. Αυτό το αρχείο μπορεί να εισαχθεί σε οποιαδήποτε Android συσκευή και να εγκατασταθεί με τον ίδιο τρόπο όπως και με άλλες εφαρμογές. Έτσι ο διαμοιρασμός είναι πολύ εύκολος και αρκείται μόνο στην διάδοση αυτού του αρχείου. Έχει οριστεί

περιγραφικό εικονίδιο και όνομα στην εφαρμογή ώστε να μπορεί ο χρήστης να την εντοπίζει και να την ανοίγει κάθε φορά που θέλει να χρησιμοποιήσει το σύστημα. Η μόνη ρύθμιση που πρέπει να γίνεται από πριν είναι η ενεργοποίηση του Bluetooth στην συσκευή του χρήστη.

## 5 ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : Εφαρμογή Υπολογιστή

Ένας από τους πιο σημαντικούς λόγους όπου επιλέχθηκε η Unity για την ανάπτυξη της εφαρμογής, είναι γιατί προσδίδει την ικανότητα εισαγωγής του 3D μοντέλου του συστήματος στην εφαρμογή και μπορεί να προσομοιώσει καταστάσεις φυσικής. Στην παρούσα εργασία δεν χρησιμοποιήθηκε περιβάλλον φυσικής προσομοίωσης, όμως κάποιος μπορεί πολύ εύκολα, και με το 3D μοντέλο που υπάρχει, να κάνει μαθηματική μοντελοποίηση του συστήματος και έτσι να κάνει δοκιμές ελεγκτών επάνω στο εικονικό σύστημα και όχι στο πραγματικό. Σε αυτό το κεφάλαιο θα περιγραφεί η πορεία για την δημιουργία του 3D μοντέλου, η εισαγωγή του στο Unity και ο τρόπος που χρησιμοποιείται. Ακόμη, η ανάπτυξη του κώδικα για την σύνδεση με το πραγματικό σύστημα και το UI της συγκεκριμένης εφαρμογής καθώς και τι προσφέρει στον χρήστη.

Ο χώρος εργασίας του Unity αποτελείται από τέσσερις βασικούς χώρους. Αριστερά είναι ο χώρος όπου τοποθετούνται όλα τα αντικείμενα της εφαρμογής σε ιεραρχία, σε αυτό το χώρο θα τοποθετηθεί το μοντέλο και τα UI στοιχεία. Στην μέση υπάρχει ο χώρος σκηνής όπου σε αυτόν τοποθετούνται τα προηγούμενα αντικείμενα στον τρισδιάστατο χώρο. Από τον ίδιο χώρο μπορεί να γίνει και προβολή της εφαρμογής όσο αυτή εκτελείται. Δεξιά σε στήλη εμφανίζονται όλες οι πιθανές ρυθμίσεις που μπορεί να έχει ένα επιλεγμένο αντικείμενο. Οι ρυθμίσεις που εμφανίζονται εξαρτώνται από τον τύπο του αντικειμένου και συνήθως έχουν να κάνουν με την θέση του στον χώρο και ειδικές ρυθμίσεις για το καθένα. Σε αυτό το σημείο τοποθετούνται και νέες ρυθμίσεις μέσω script όπου ελέγχουν το αντικείμενο μέσα από κώδικα. Στο κάτω μέρος του παραθύρου υπάρχει η δομή αρχείων του project και οργανωμένα σε φακέλους υπάρχουν τα μοντέλα που χρησιμοποιούνται, τα script που έχουν τον κώδικα για την λειτουργία της εφαρμογής και υφές (textures) όπου χρησιμοποιούνται για τον χρωματισμό και την επιλογή υλικών που θα έχουν τα αντικείμενα.



Εικόνα 5.1: Το περιβάλλον του προγράμματος Unity

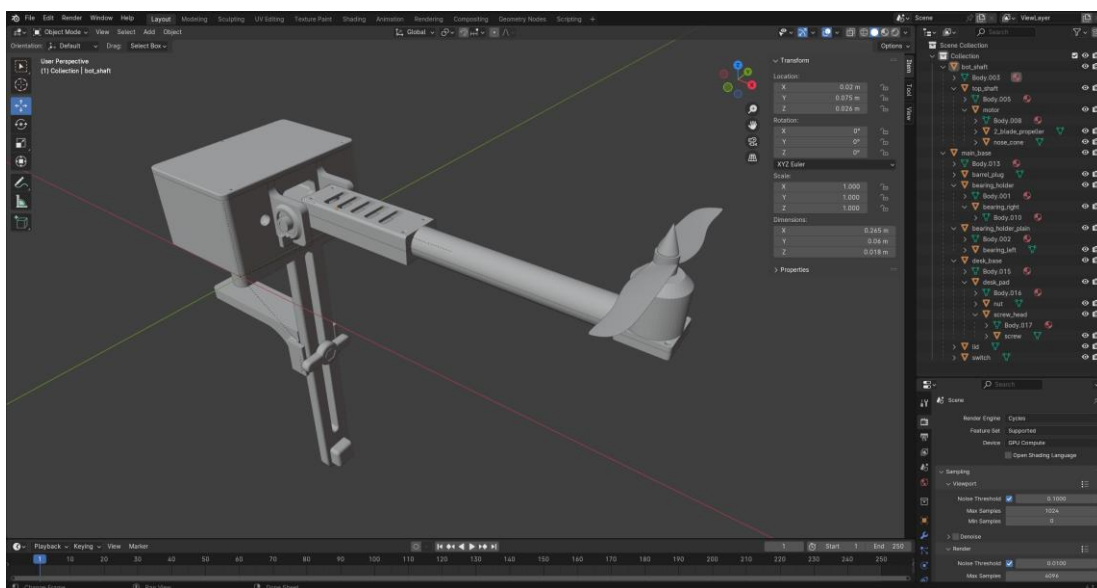
Ο κώδικας γράφεται σε γλώσσα C# και χρησιμοποιείται ο text editor του Visual Studio της Microsoft για την συγγραφή του. Χωρίζεται σε scripts τα οποία τοποθετούνται σε αντικείμενα και συνήθως το κάθε script εκτελεί μία συγκεκριμένη διεργασία στο αντικείμενο. Για κάθε συλλογή UI αντικειμένων έχει δημιουργηθεί αντίστοιχο script όπου εφαρμόζει την απαραίτητη λογική για την λειτουργία του κάθε στοιχείου.

## 5.1 3D Μοντέλα στο Unity και Blender

Για το 3D μοντέλο απαιτείται ο σχεδιασμός του σε πρόγραμμα και μετά η εισαγωγή του στο Unity. Εφόσον σε προηγούμενα βήματα, κατά την κατασκευή του συστήματος, έχουν δημιουργηθεί ήδη 3D μοντέλα για την 3D εκτύπωση, θα χρησιμοποιηθούν, ενώ τα υπόλοιπα εξαρτήματα θα χρειαστεί να σχεδιαστούν. Για την σύνδεσή τους σε ένα ενιαίο μοντέλο χρησιμοποιήθηκε το Blender. Έγινε εισαγωγή των μεμονωμένων μοντέλων και ιεραρχήθηκαν με βάση την θέση τους και την σύνδεση που παρουσιάζουν με τα γειτονικά τους εξαρτήματα. Πέρα από το μοντέλο του συστήματος, δημιουργήθηκε και μοντέλο ενός εικονικού δωματίου για την ενίσχυση της εμπειρίας του χρήστη.

### 5.1.1 Μοντέλο Συστήματος

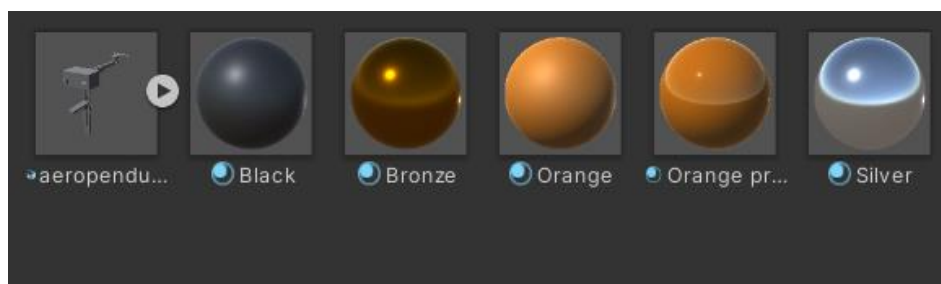
Με το FreeCAD σχεδιάστηκαν μοντέλα για τις βίδες, τα ρουλεμάν και τον κινητήρα. Για τον έλικα, βρέθηκε το 3D μοντέλο του online στην σελίδα thingiverse.com. Η σελίδα αυτή διαθέτει δωρεάν 3D μοντέλα κυρίως για 3D εκτύπωση σχεδιασμένα από χρήστες της σελίδας (Thingiverse, χ.χ.). Έχοντας έτσι όλα τα μοντέλα στο FreeCAD, έγινε εξαγωγή τους (export) σε .obj αρχεία όπου μπορούν να επεξεργαστούν από το Blender. Φορτώνοντας όλα τα αρχεία στο περιβάλλον του Blender, ακολούθησε η τοποθέτησή τους σε ιεραρχία και στην κατάλληλη θέση μέσα στον 3D χώρο, με στόχο την δημιουργία αντίγραφου του πραγματικού συστήματος.



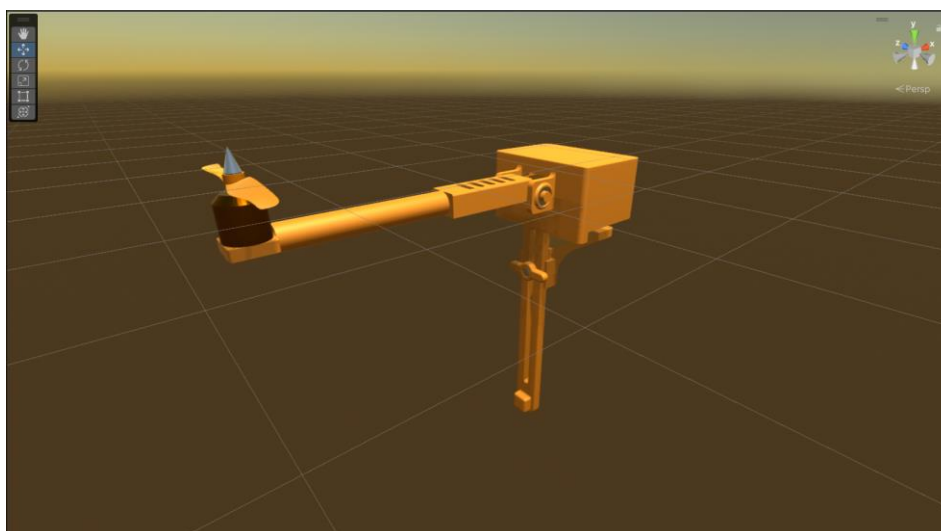
**Εικόνα 5.2: 3D αναπαράσταση του συστήματος στο περιβάλλον του Blender, με την ιεραρχία των μοντέλων να φαίνεται δεξιά**

Όπως φαίνεται στα δεξιά της παραπάνω εικόνας (Εικόνα 5.2), η ιεραρχία των μοντέλων είναι με τέτοιο τρόπο δομημένη ώστε το κάθε μοντέλο να ανήκει σε μοντέλο που είναι συνδεδεμένο το ίδιο. Η ιεραρχία αυτή θα βοηθήσει αργότερα, στον προγραμματισμό της συμπεριφοράς τους στο Unity και στον διαχωρισμό των μοντέλων. Για την τοποθέτησή τους στις σωστές θέσεις, χρησιμοποιήθηκε το μονε εργαλείο του Blender το οποίο επιτρέπει την ακριβή κίνησή τους και την περιστροφή τους γύρω από τους τρεις άξονες του χώρου. Ακολούθησε export του τελικού μοντέλου σε αρχείο .fbx ώστε να γίνει μετά η εισαγωγή τους (import) στο Unity.

Αφού έγινε δημιουργία νέου project στο Unity, με απλή drag and drop μέθοδο, έγινε import το μοντέλο του συστήματος στο περιβάλλον του Unity. Ακολούθησε δημιουργία των texture και υλικών τα οποία φτιάχτηκαν με βάση το πραγματικό σύστημα. Τα texture αυτά τοποθετήθηκαν στα αντίστοιχα στοιχεία του μοντέλου.



**Εικόνα 5.3: Τα textures και το μοντέλο του συστήματος**



**Εικόνα 5.4: Το 3D μοντέλο του συστήματος με τα textures στα σωστά αντικείμενα**

Το πραγματικό σύστημα έχει ένα κινητό σημείο, τον βραχίονα, όπου πρέπει να αναδημιουργηθεί και στο εικονικό περιβάλλον. Με την επιλογή του βραχίονα από την ιεραρχία, δημιουργήθηκε ένα νέο αντικείμενο τύπου Hinge Joint, στο οποίο θέτοντας τον σωστό άξονα περιστροφής και ένα σταθερό σημείο αναφοράς, επιτρέπει στον βραχίονα να κινηθεί ως προς την σταθερή βάση όπως και το πραγματικό. Έτσι τροποποιώντας μόνο την τιμή της γωνίας στο Hinge Joint γίνεται αυτόματα η κίνηση της περιστροφής. Ακόμη, κατά την λειτουργία του συστήματος υπάρχει script για τον έλικα όπου του επιτρέπει να περιστρέφεται όπως και στο πραγματικό σύστημα με ανάλογη ταχύτητα. Το 3D μοντέλο του συστήματος έτσι έχει ολοκληρωθεί.

### **5.1.2 Μοντέλο Περιβάλλοντος Χώρου**

Είναι σημαντικό για τον χρήστη να έχει θετική εμπειρία από την εφαρμογή κατά την χρήση της. Με την εισαγωγή ενός εικονικού δωματίου, μέσα στο οποίο βρίσκεται το σύστημα, προσφέρεται οικειότητα με τον χώρο. Έτσι βρέθηκε online ένα μοντέλο γραφείου (αυτή τη φορά από το sketchfab.com). Με την εισαγωγή του στο Blender έγιναν μικρές διορθώσεις και δημιουργήθηκαν εικονικοί τοίχοι ώστε το περιβάλλον να θυμίζει δωμάτιο.



**Εικόνα 5.5: Το μοντέλο γραφείου και περιβάλλοντος χώρου στο Blender**

Ακολουθώντας την ίδια διαδικασία με το μοντέλο του συστήματος, έγινε η εισαγωγή του στο Unity. Αυτή τη φορά το μοντέλο του γραφείου είχε ήδη textures και μόνο οι τοίχοι χρειάστηκαν βάψιμο. Το μοντέλο του συστήματος τοποθετήθηκε σε τέτοια θέση ώστε η βάση στήριξης του να είναι πιασμένη στο εικονικό γραφείο, κρατώντας έτσι στοιχεία ρεαλισμού. Ένα πολύ βασικό χαρακτηριστικό του χώρου είναι ο φωτισμός, ο οποίος έγινε με την εισαγωγή εικονικής πηγής φωτός στο επάνω μέρος του δωματίου και ρυθμίστηκε η έντασή του ώστε να θυμίζει πραγματικό δωμάτιο.

Για την ενίσχυση του ρεαλισμού στο χώρο, έγινε εισαγωγή ενός service από το ενσωματωμένο Package Manager του Unity, το Post Processing, όπου διαθέτει μια γκάμα ρυθμίσεων για εφέ και φωτισμό στον χώρο. Μερικές από τις ρυθμίσεις αυτές είναι επιγραμματικά: Bloom, Motion Blur και Ambient Occlusion. Προστέθηκαν επίσης σκιές οι οποίες δημιουργούνται από την παρουσία του φωτός στο χώρο και σχεδιάζονται σε πραγματικό χρόνο κατά τη λειτουργία της εφαρμογής.

Άλλο ένα πολύ βασικό αντικείμενο είναι η κάμερα. Πρόκειται για αντικείμενο όπου προσομοιώνει μία κάμερα στον 3D χώρο μέσα από την οποία εμφανίζονται όλα τα στοιχεία στην εφαρμογή. Τοποθετήθηκε σε σημείο τέτοιο ώστε το σύστημα να είναι εντός πλάνου. Προκειμένου να ενισχυθεί η οπτική εμπειρία του χρήστη, γράφηκε script όπου με την χρήση του ποντικιού ή συγκεκριμένων κουμπιών στο πληκτρολόγιο μπορεί να γίνει περιστροφή γύρω από το σύστημα καθώς και μεγέθυνση (zoom). Το script αυτό τοποθετήθηκε στο αντικείμενο της κάμερας.

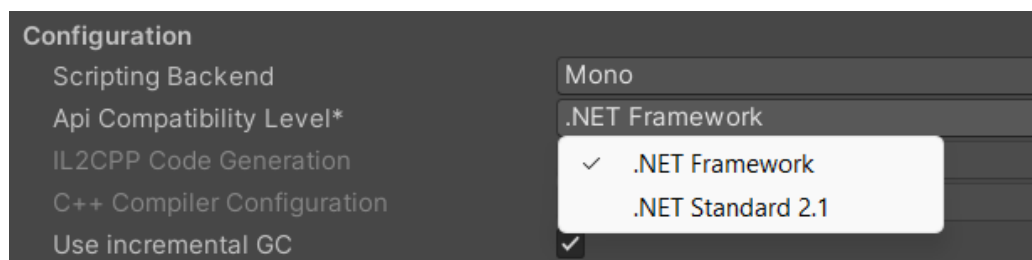




**Εικόνα 5.6:** Το εικονικό δωμάτιο με την κάμερα, την πηγή φωτισμού και το σύστημα στερεωμένο στο γραφείο

## 5.2 Σειριακή επικοινωνία

Για την επικοινωνία του συστήματος με το Unity, έγινε χρήση κατάλληλης βιβλιοθήκης στο Unity η οποία επιτρέπει την επικοινωνία του προγράμματος με εισόδους και εξόδους το ηλεκτρονικού υπολογιστή. Η βιβλιοθήκη είναι η System.IO.Ports και προκειμένου να χρησιμοποιηθεί απαιτείται η αλλαγή μιας ρύθμισης στα Project Settings, η αλλαγή της χρήσης του ‘.NET Standard’ σε ‘.NET Framework’ για το API Compatibility Level. Η αλλαγή αυτή είναι ζωτικής σημασίας καθώς η Standard έκδοση δεν μπορεί να έχει πρόσβαση στις περιφερειακές συσκευές του υπολογιστή.



**Εικόνα 5.7:** Αλλαγή ρύθμισης του API Compatibility Level από τα Project Settings

Στο script με όνομα Serial Communication υπάρχουν εντολές και συναρτήσεις για την σύνδεση με το ESP32 και για την επικοινωνία τους με τα κωδικοποιημένα string. Όπως και στην εφαρμογή κινητού, έχει γραφεί ειδικό script υπεύθυνο για την αποκωδικοποίηση της εισερχόμενης πληροφορίας και κωδικοποίηση της εξερχόμενης, μέσω συναρτήσεων.

Με την λήψη νέων δεδομένων, αποκρυπτογραφούνται και αποθηκεύονται σε μεταβλητές ώστε να χρησιμοποιηθούν από άλλα σημεία του κώδικα. Αρχικά ενημερώνεται το στοιχείο UI που είναι υπεύθυνο για την προβολή της κατάστασης του συστήματος, αναγράφοντας την γωνία, την ταχύτητα και την εκτελούμενη λειτουργία. Ακόμη, με τις τιμές της γωνίας ενημερώνεται και το 3D μοντέλο σε πραγματικό χρόνο αποδίδοντας την ίδια γωνία με το πραγματικό σύστημα. Η τιμή της ταχύτητας του

κινητήρα επίσης αποτυπώνεται στην ταχύτητα περιστροφής του έλικα στο μοντέλο. Τέλος, όπως και στην mobile εφαρμογή, για λόγους συμβατότητας, η τιμή του state ορίζει το προβαλλόμενο παράθυρο ανάλογα με την λειτουργία που εκτελεί το ESP32.

Κατά την αποστολή δεδομένων, πατώντας το κουμπί start/stop στην εφαρμογή, αποστέλλεται πληροφορία στο ESP32 ανάλογα με την λειτουργία που έχει επιλεγεί και τις τιμές από τα αντίστοιχα πεδία για τις παραμέτρους του PID.

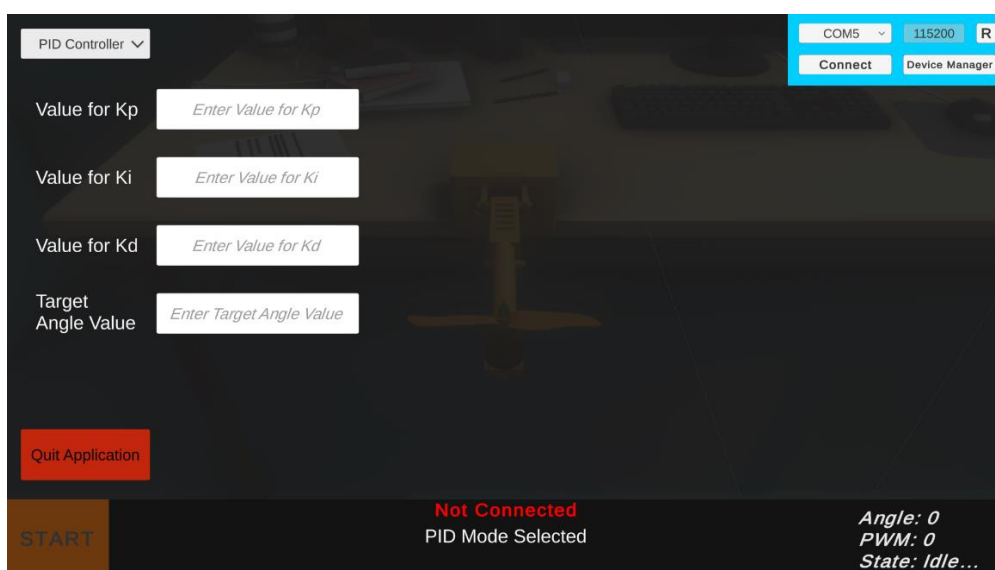
### 5.3 Unity User Interface

Όπως προαναφέρθηκε, το UI θυμίζει αυτό της mobile εφαρμογής ως προς την λογική, με προφανώς διαφορετική δομή ώστε να ταιριάζει στον διαφορετικό τρόπο ελέγχου από το χρήστη. Υπάρχει ένα dropdown μενού στο επάνω μέρος της οθόνης όπου γίνεται η επιλογή της λειτουργίας που επιθυμεί να εκτελέσει ο χρήστης. Επιλέγοντας διαφορετική λειτουργία ενεργοποιούνται αντίστοιχα παράθυρα για ρυθμίσεις παραμέτρων. Ένα παράθυρο το οποίο είναι μόνιμα ενεργό κατά την ρύθμιση είναι αυτό της επιλογής COM port και σύνδεσης με το ESP32.

Με την εκτέλεση της εφαρμογής, εμφανίζεται ένα πάνελ ρυθμίσεων που προσφέρει:

1. επιλογή λειτουργίας ανάμεσα σε τρεις: την χειροκίνητη λειτουργία, την λειτουργία PID και μία νέα λειτουργία, την live view κατά την οποία απλά εμφανίζεται το 3D μοντέλο του συστήματος στο χώρο και παρουσιάζεται η σύνδεση ανάμεσα στο πραγματικό σύστημα με το εικονικό.
2. Την ρύθμιση των παραμέτρων του PID ελεγκτή όταν έχει επιλεγεί αυτή η λειτουργία
3. Η επιλογή της COM port στην οποία θα κάνει προσπάθεια σύνδεσης η εφαρμογή, προκειμένου να συνδεθεί με το ESP32

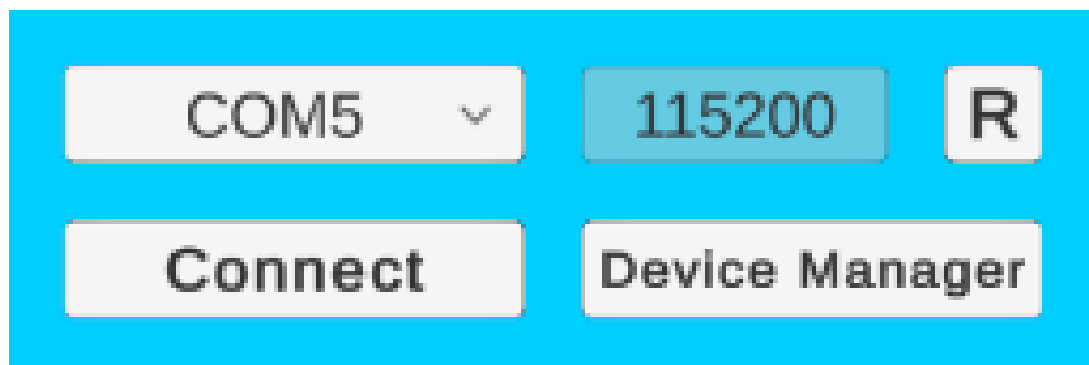
Κατά την εκτέλεση κάποιας από τις λειτουργίες, εμφανίζεται το εικονικό μοντέλο του συστήματος και ενεργοποιούνται νέες ρυθμίσεις όπως για παράδειγμα η ρύθμιση ταχύτητας κατά την χειροκίνητη λειτουργία.



Εικόνα 5.8: Η οθόνη των ρυθμίσεων, το πάνελ σύνδεσης επάνω δεξιά με μπλε χρώμα, το dropdown επιλογής λειτουργίας επάνω αριστερά και η μπάρα πληροφοριών στο κάτω μέρος

### 5.3.1 COM port ρυθμίσεις

Το πάνελ σύνδεσης με το ESP32 το οποίο βρίσκεται επάνω στα δεξιά της οθόνης ρυθμίσεων, αποτελείται από ένα dropdown μενού από το οποίο επιλέγεται η κατάλληλη συσκευή, και τρία κουμπιά: το κουμπί σύνδεσης, ένα κουμπί για να γίνεται ανανέωση των συσκευών και ένα για να ανοίγει την Διαχείριση Συσκευών στα windows.



Εικόνα 5.9: Το πάνελ σύνδεσης COM port

Οι επιλογές του dropdown αντιπροσωπεύουν τις σειριακά συνδεδεμένες συσκευές στον ηλεκτρονικό υπολογιστή την συγκεκριμένη χρονική στιγμή. Σε περίπτωση που ο χρήστης ανοίξει την εφαρμογή πριν συνδέσει το ESP32, δεν θα εμφανιστεί η σωστή COM port, έτσι με το κουμπί ανανέωσης (refresh) ανανεώνονται οι τιμές με τις νέες συσκευές. Για την διευκόλυνση του χρήστη, το κουμπί Device Manager ανοίγει την Διαχείριση Συσκευών ώστε να μπορεί με ευκολία ο χρήστης να δει ποια από τις συσκευές του είναι η σωστή ώστε να κάνει επιτυχή σύνδεση.

Όλες οι τιμές του πάνελ όπου επιλέγονται, τροποποιούν global μεταβλητές και με το πάτημα του κουμπιού Connect εκτελείται η συνάρτηση σύνδεσης στο Serial Communication Script με τις σωστές παραμέτρους. Σε αυτό το στάδιο, η επιλογή του baud rate είναι κλειδωμένη στα 115200 όπου και είναι η πιο συνηθισμένη τιμή για το ESP32. Σε προσπάθεια που έγινε για να μπορεί να ρυθμιστεί από τον χρήστη, η εφαρμογή δεν ανταποκρινόταν αν δεν ήταν σωστή η τιμή και έτσι αποφασίστηκε να κλειδωθεί. Ο κώδικας για την τροποποίησή του υπάρχει και μπορεί να ενεργοποιηθεί από χρήστες οι οποίοι έχουν κατάλληλες γνώσεις.

### 5.3.2 Ρυθμίσεις PID

Επιλέγοντας την λειτουργία PID από το dropdown εμφανίζονται τέσσερα πεδία. Σε αυτά τα πεδία ο χρήστης μπορεί να βάλει τις παραμέτρους για τον PID ελεγκτή και την επιθυμητή γωνία σταθεροποίησης. Με ειδικό script γίνεται έλεγχος των τιμών ώστε να πληρούν κάποιες προϋποθέσεις και να είναι εντός ορίων.

Εικόνα 5.10: Οι ρυθμίσεις για την PID λειτουργία

### 5.3.3 Μπάρα πληροφοριών και ελέγχου

Σε όποιο στάδιο ρυθμίσεων ή λειτουργιών και να βρίσκεται η εφαρμογή, η μπάρα στο κάτω μέρος προβάλλεται μονίμως. Στα αριστερά της μπάρας υπάρχει το κουμπί START το οποίο ενεργοποιεί ή απενεργοποιεί την εκάστοτε λειτουργία. Στα δεξιά υπάρχει κείμενο όπου προβάλλει την κατάσταση του συστήματος και ενημερώνεται κάθε φορά που το ESP32 αποστέλλει νέα πληροφορία.

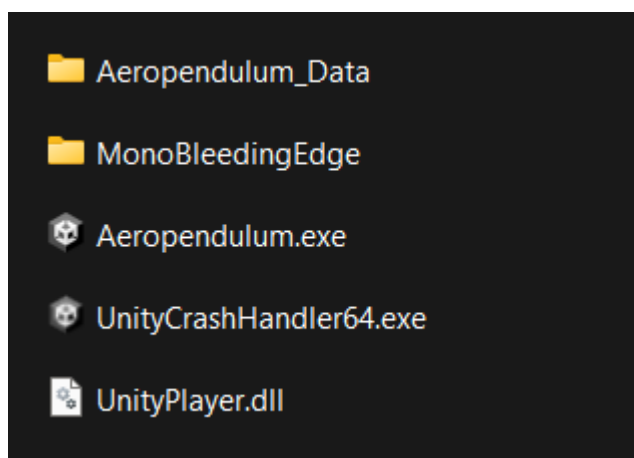
Στις ρυθμίσεις, η μέση της μπάρας προβάλλει την κατάσταση σύνδεσης με το ESP32, την λειτουργία που έχει επιλεγεί και μηνύματα σφάλματος τα οποία προειδοποιούν τον χρήστη. Στην λειτουργία για τον PID προβάλλονται οι παράμετροι που εφαρμόστηκαν, ενώ στο manual mode υπάρχει slider ώστε να επιλέγεται η ταχύτητα του κινητήρα και νέο κουμπί ώστε να αποστέλλεται αυτή στο ESP32.

START	<p style="color: green;">System Connected</p> <p>Live View Mode Selected</p>		<p>Angle: -90</p> <p>PWM: 0</p> <p>State: Idle...</p>
START	<p>● ————— 0</p>	<p>Apply</p>	<p>Angle: -90</p> <p>PWM: 0</p> <p>State: Manual</p>
START	<p>P: 0</p> <p>I: 0</p> <p>D: 0</p>	<p>Target Angle: 0 Degrees</p>	<p>Angle: -90</p> <p>PWM: 0</p> <p>State: Pid</p>

Εικόνα 5.11: Η μπάρα στις τρεις καταστάσεις του συστήματος. Επάνω στην κατάσταση Idle, στη μέση στην χειροκίνητη λειτουργία με το slider και το κουμπί εφαρμογής και κάτω στην PID λειτουργία με την προβολή των παραμέτρων και τις γωνίας σταθεροποίησης

## 5.4 Εφαρμογή σε Unity

Προκειμένου ο χρήστης να μπορεί να χρησιμοποιήσει την εφαρμογή αυτή, πρέπει να δημιουργηθεί το εκτελέσιμο .exe αρχείο. Πριν γίνει αυτό, έγινε ρύθμιση των γραφικών από τα Project Settings και ρυθμίστηκαν στις μέγιστες τιμές τους. Παρόλο που αυτό μπορεί να επιβαρύνει κάποιον πολύ αδύναμο υπολογιστή, τα μοντέλα και οι ρυθμίσεις δεν επιβαρύνουν καθόλου κάποιο σύγχρονο μηχάνημα, αφού υπάρχουν ελάχιστα αντικείμενα σε σχέση με ένα αντίστοιχο ηλεκτρονικό παιχνίδι. Για την δημιουργία του εκτελέσιμου αρχείου από το File / Build Settings, αφού επιλέχθηκαν τα Windows ως πλατφόρμα στόχος, με το Build κουμπί έγιναν εξαγωγή σε φάκελο όλα τα αρχεία συμπεριλαμβανόμενου και του .exe. Με την διάδοση αυτού μόνο του φακέλου σε χρήστες, μπορεί να γίνει η εκτέλεση της εφαρμογής ανεξαρτήτως χαρακτηριστικών συστήματος, μόνο με το άνοιγμα του .exe αρχείου.



Εικόνα 5.12: Τα αρχεία που δημιουργήθηκαν μετά το Build της εφαρμογής

## 6 ΚΕΦΑΛΑΙΟ 6<sup>ο</sup> : Διανομή Ανοικτού Κώδικα

Με το σύστημα ολοκληρωμένο και τις εφαρμογές έτοιμες για χρήση, αυτό που απέμεινε είναι ένας τρόπος διάδοσης του πακέτου σε ενδιαφερόμενους σε όλον τον κόσμο. Με τη χρήση του GitHub αυτό μπορεί να γίνει εφικτό. Μιας και ο στόχος αυτής της εργασίας είναι η διάδοση του συστήματος σε άτομα διαφορετικών ικανοτήτων και γνώσεων, είναι υποχρεωτική η γραφή κειμένων που περιγράφουν τις λειτουργίες του συστήματος όπως και αναλυτικό documentation στα Αγγλικά. Οδηγίες χρήσης και κατασκευής όπως και λίστα εξαρτημάτων είναι επίσης απαραίτητα. Σε αυτό το κεφάλαιο θα παρουσιαστούν τα βήματα που έγιναν για την επίτευξη των παραπάνω, από την γραφή οδηγιών μέχρι και το ανέβασμα στο αποθετήριο GitHub.

### 6.1 Οδηγίες Χρήσης

Προκειμένου να διευκολυνθούν οι χρήστες οι οποίοι θέλουν απλά να αναδημιουργήσουν το σύστημα και να χρησιμοποιήσουν τις παρεχόμενες εφαρμογές, έγινε δημιουργία animation μικρού μήκους και βίντεο με οδηγίες λειτουργίας ώστε να μην μπουν σε διαδικασία εκτενούς διαβάσματος του documentation. Προφανώς, οι περισσότερες πληροφορίες και λεπτομέρειες θα βρίσκονται στις γραπτές οδηγίες. Για άτομα όπου θέλουν να επεκταθούν πέρα από το υλικό που παρέχεται ή να το βελτιώσουν, υπάρχουν σχόλια σε όλον τον κώδικα και σε όλες τις εφαρμογές, ο οποίος περιγράφει συνοπτικά την λειτουργία των συναρτήσεων που έχουν γραφεί.

#### 6.1.1 Γραπτές Οδηγίες

Για τις γραπτές οδηγίες, αποφασίστηκε ως γλώσσα γραφής να είναι τα Αγγλικά μιας και το πακέτο δεν προορίζεται αποκλειστικά για ελληνικό κοινό. Κατά τη συγγραφή των οδηγιών δόθηκε αρκετή προσοχή στην σωστή τεκμηρίωση των αποφάσεων που πάρθηκαν ώστε να μπορέσει ο καθένας να κρίνει αν κάποια επιλογή είναι σωστή και για τον ίδιο. Η δομή του documentation αρχείου είναι η εξής:

- Σύντομη περιγραφή του συστήματος και των δυνατοτήτων του και σύνδεσμος για το βίντεο λειτουργίας.
- Περιγραφή της δομής του πακέτου, που μπορούν να βρεθούν όλα τα αρχεία.
- Λίστα εξαρτημάτων με αναλυτική περιγραφή όπως και τα αντίστοιχα datasheets τους.
- Σύντομες οδηγίες κατασκευής και σύνδεσμος για το βίντεο κατασκευής.
- Απλά παραδείγματα σύνδεσης του συστήματος με τις εφαρμογές τους και τον τρόπο χρήσης τους.
- Ειδικές περιπτώσεις για πιθανές αλλαγές στις εφαρμογές ή στο σύστημα που μπορούν να εφαρμόσουν οι έμπειροι χρήστες.

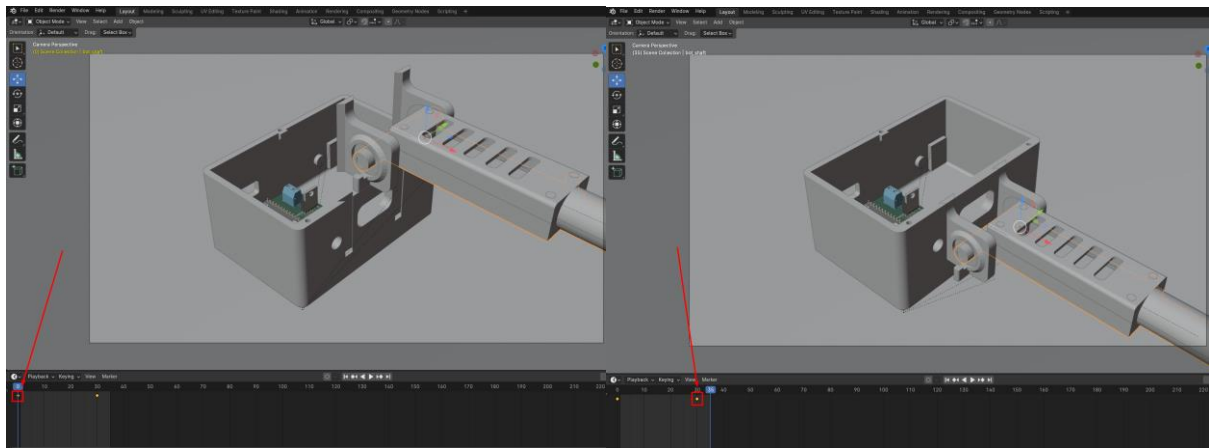
Με βάση το παραπάνω πλάνο γράφηκε χωρίς μεγάλη δυσκολία το documentation και αποθηκεύτηκε στον φάκελο του πακέτου.

#### 6.1.2 Οδηγίες σε Animation

Δημιουργήθηκε animation μικρού μήκους το οποίο παρουσιάζει τις οδηγίες κατασκευής του συστήματος βήμα-βήμα. Η διαδικασία αυτή ήταν εύκολη μιας και υπήρχε ήδη το μοντέλο του

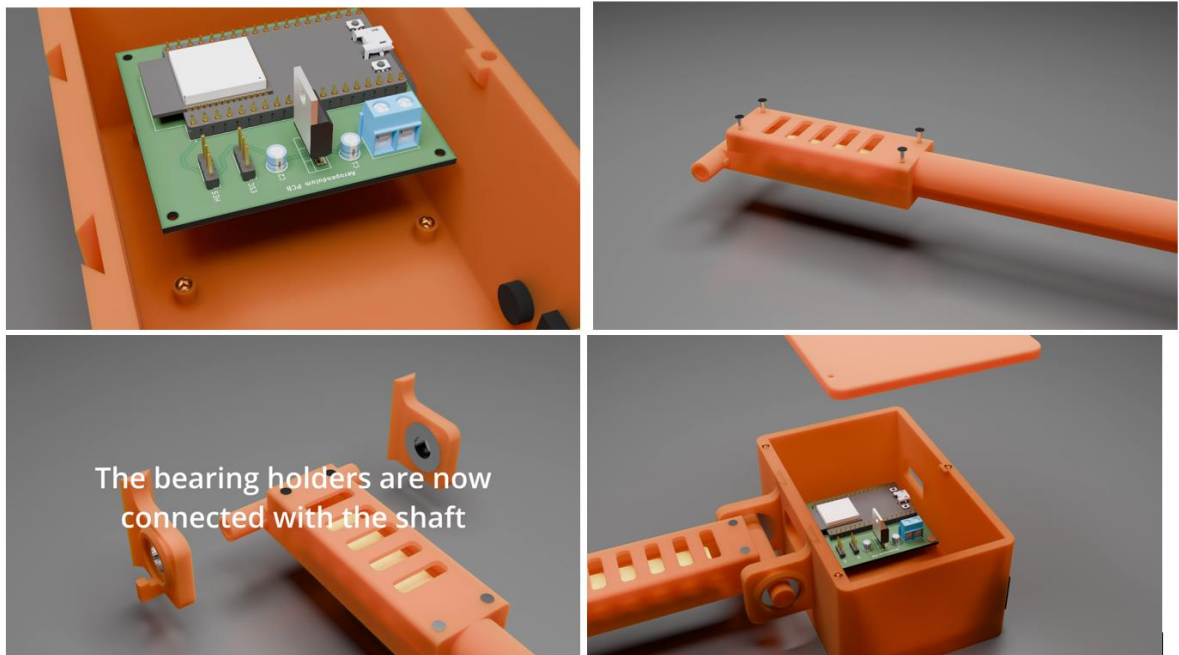
συστήματος σε κομμάτια. Έτσι με το Blender έγινε σκηνοθεσία των animation όπου θα φτιαχτούν, τοποθετώντας κατάλληλο φωτισμό επάνω από το μοντέλο.

Η δημιουργία των σκηνών έγινε με την χρήση keyframe του Blender, κατά την οποία θέτεται μια αρχική θέση ενός αντικειμένου στο χώρο και μια τελική. Το Blender έτσι, αφού έχει οριστεί κατάλληλη διαφορά χρόνου από την αρχική στην τελική θέση, κάνει animate την κίνηση του μοντέλου ένα καρέ (frame) τη φορά και τα εξάγει σε μορφή πολλών φωτογραφιών (μία για κάθε frame). Ανάλογα με τα frames per second (FPS) που θα έχει το τελικό βίντεο, γίνεται γρήγορη εναλλαγή των εικόνων αυτών σε αλληλουχία και έτσι δημιουργείται η ψευδαίσθηση της κίνησης. Δημιουργήθηκαν εν τέλη πάνω από είκοσι τέτοιες σκηνές, οι οποίες συνδυάστηκαν σε πρόγραμμα επεξεργασίας βίντεο.



**Εικόνα 6.1:** Περιγραφή ενός keyframe, αριστερά ορίζεται η αρχική θέση του βραχίονα στον χώρο σε συγκεκριμένο χρόνο, δεξιά η τελική του θέση στον χώρο και χρόνο

Το πρόγραμμα επεξεργασίας βίντεο είναι το Davinci Resolve, στο οποίο αφού τοποθετήθηκαν όλες οι σκηνές στην χρονογραμμή του, έγινε πρόσθεση υποτίτλων με τις οδηγίες σε γραπτή μορφή και παύσεων ώστε να μπορεί ο κάθε χρήστης να διαβάσει και να επεξεργαστεί το κάθε βήμα της κατασκευής. Τέλος έγινε εξαγωγή του βίντεο σε αρχείο .mp4 και ανέβηκε στο YouTube ώστε να είναι διαθέσιμο σε όλους χωρίς την επιβάρυνση του χώρου που μπορεί να καταλαμβάνει.

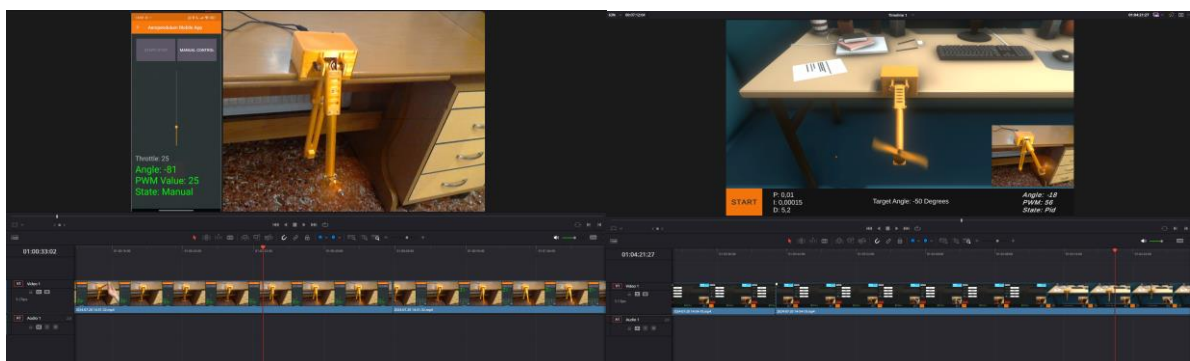


Εικόνα 6.2: Μερικές εικόνες από το τελικό βίντεο animation

### 6.1.3 Βίντεο Παρουσίασης Λειτουργιών

Η διαδικασία δημιουργίας του βίντεο περιγραφής των λειτουργιών χωρίζεται σε δύο κομμάτια. Το πρώτο κομμάτι είναι η βιντεοσκόπηση του συστήματος και καταγραφή των εφαρμογών την ίδια στιγμή, ώστε οι κινήσεις στην εφαρμογή να αντιστοιχούν σε κινήσεις του συστήματος και αντίστροφα. Αυτό έγινε με την βοήθεια κάμερας από το PC και ταυτόχρονα χρήση προγραμμάτων καταγραφής οθόνης για τις εφαρμογές. Στην διάρκεια της καταγραφής παρουσιάστηκαν όλες οι λειτουργίες που μπορεί να εκτελέσει το σύστημα.

Το δεύτερο βήμα είναι η επεξεργασία των παραπάνω βίντεο ώστε να γίνουν ένα ενιαίο. Με την βοήθεια πάλι του Davinci Resolve, έγινε επεξεργασία των βίντεο ώστε να συγχρονιστούν η βιντεοσκόπηση και η καταγραφή και να τοποθετηθούν με τέτοιο τρόπο ώστε να είναι ευδιάκριτες οι κινήσεις που έγιναν.



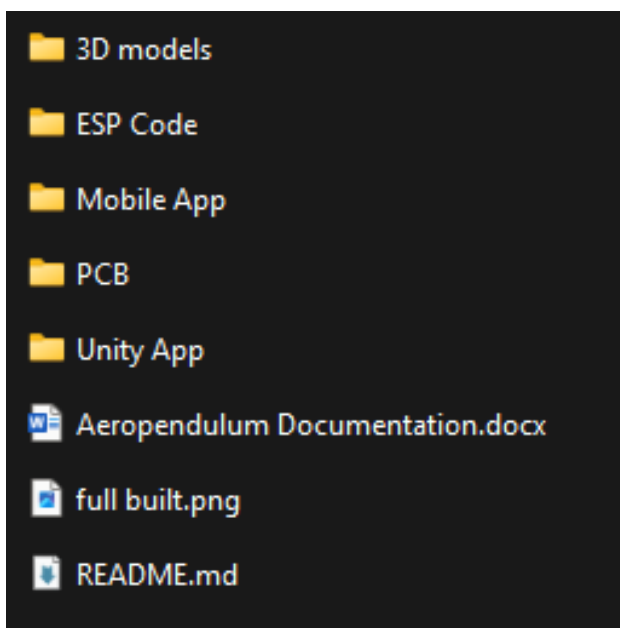
Εικόνα 6.3: Αποκόμματα κατά την επεξεργασία των βίντεο με το Davinci Resolve. Αριστερά η χρήση της mobile εφαρμογής, δεξιά χρήση εφαρμογής στο Unity.



## 6.2 Οργάνωση αρχείων σε Online Αποθετήριο

Έχοντας όλα τα αρχεία και τις οδηγίες έτοιμα, έγινε οργάνωσή τους σε φακέλους ανάλογα με το κομμάτι στο οποίο ανήκει κάθε αρχείο. Ότι έχει να κάνει με τα αρχεία που σχεδιάστηκαν με 3D πρόγραμμα, είτε αυτά εκτυπώθηκαν είτε χρησιμοποιήθηκαν στο Unity ή στο Animation, υπάρχουν στον φάκελο 3D models. Τα αρχεία για το PCB από το KiCAD όπως και τα αρχεία για την κατασκευή του PCB (τα Gerbers) υπάρχουν στον φάκελο PCB.

Όσον αφορά τον κώδικα, το λογισμικό του ESP32 βρίσκεται στον φάκελο ESP Code και των δύο άλλων εφαρμογών στα Mobile App και Unity App αντίστοιχα, μαζί με τα εκτελέσιμα αρχεία τους. Στον φάκελο του πακέτου έχουν τοποθετηθεί επίσης φωτογραφία του συστήματος από το μοντέλο στο Blender, README αρχείο καθώς και το documentation.



Εικόνα 6.4: Ο φάκελος του πακέτου με όλα τα αρχεία οργανωμένα σε υποφακέλους

Τα αρχεία αυτά θα ανέβουν τελικά στο GitHub και έτσι ο καθένας θα μπορεί να έχει πρόσβαση σε αυτά ώστε να τα χρησιμοποιήσει όπως επιθυμεί. Είτε για να χρησιμοποιήσει το σύστημα, να το βελτιώσει και να το επεκτείνει, να διορθώσει πιθανά λάθη, να δανειστεί αρχεία ή κώδικα και πολλά άλλα. Η φαντασία και η δημιουργικότητα του καθενός θα είναι το όριο αυτών που μπορούν να γίνουν.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτό το τελευταίο κεφάλαιο θα γίνει παρουσίαση των στόχων της εργασίας και πως αυτοί επιτεύχθηκαν. Ακόμη θα παρουσιαστούν οι τρόποι με τους οποίους συνεισφέρει η εργασία αυτή στο πεδίο των εκπαιδευτικών εργαλείων και της open source κοινότητας. Θα αναλυθούν προβλήματα που παρουσιάστηκαν κατά την πορεία της εργασίας και αν ή με ποιους τρόπους ξεπεράστηκαν. Τέλος, θα γίνει κατανοητό πως αδυναμίες του παρόντος εργαλείου μπορούν να εκμεταλλευτούν από άλλους για την βελτίωσή του ή την επέκτασή του σε νέα πεδία της επιστήμης, όπως και να συζητηθούν ιδέες που, λόγω χρόνου ή δυσκολίας, δεν εφαρμόστηκαν στο συγκεκριμένο σύστημα.

Όπως είναι εμφανές, επιτεύχθηκε η δημιουργία ενός εκπαιδευτικού πακέτου εκμάθησης και εξάσκησης με PID ελεγκτές. Στον χρόνο που κράτησε η διαδικασία της δημιουργίας του πακέτου αυτού, ερευνήθηκαν τεχνολογίες, εξαρτήματα και εργαλεία τα οποία βοήθησαν στην δημιουργία συστημάτων και εφαρμογών. Με την ολοκλήρωση της εργασίας αυτής, άτομα από όλον τον κόσμο μπορούν να έχουν πρόσβαση στην έρευνα που έγινε γύρω από τα εκπαιδευτικά εργαλεία και τρόπους με τους οποίους μπορούν να δημιουργήσουν τα δικά τους. Αναλύθηκαν τρόποι για τον σχεδιασμό και 3D εκτύπωσης εξαρτημάτων, την κατασκευή PCB και κόλληση εξαρτημάτων πάνω τους, την εύρεση του κατάλληλου μικροελεγκτή ανάλογα το πεδίο εφαρμογής, τον προγραμματισμό τέτοιων μικροελεγκτών, της ανάπτυξης εφαρμογών και της δημιουργίας οδηγιών, μεταξύ άλλων. Με την διάδοση του πακέτου, εμπλουτίστηκε ο τομέας των open source εκπαιδευτικών εργαλείων και έγινε πιο προσβάσιμη η κατασκευή τους από άτομα διαφορετικών δυνατοτήτων.

Τα περισσότερα προβλήματα που αντιμετωπίστηκαν, έχουν αναλυθεί σε προηγούμενα κεφάλαια, όπως ο λόγος που εμφανίστηκαν και οι λύσεις τους. Υπάρχουν όμως και άλλα τα οποία παραμένουν άλυτα ή έχουν περιοριστεί με τέτοιον τρόπο ώστε να μην επηρεάζουν την λειτουργία του πακέτου.

Όσον αφορά το σύστημα, ο κινητήρας που αντικαταστάθηκε, εξακολουθεί να είναι ισχυρός και παράγει πολύ παραπάνω δύναμη για το βάρος του βραχίονα. Αυτό έχει ως αποτέλεσμα την απότομη αλλαγή από βαθμίδα σε βαθμίδα ταχύτητας ελέγχου, κάνοντας το σύστημα να αντιδρά με υπερβολικό τρόπο μερικές φορές χωρίς να φταίει όμως η ρύθμιση του PID ελεγκτή. Ακόμη, υπάρχει μεγάλη ανακρίβεια στην ανίχνευση της γωνίας του βραχίονα λόγω της δομής του διαμετρικού μαγνήτη και της ύπαρξης ενός μόνο hall effect αισθητήρα όπου μπορεί να ανιχνεύσει λανθασμένα. Λύση σε αυτό το πρόβλημα δεν δόθηκε, όμως έγινε έρευνα γύρω από παρόμοιους τρόπους ανίχνευσης γωνίας, με χρήση δύο hall effect αισθητήρων όπου βρίσκονται σε γωνία 90 μοιρών ο ένας με τον άλλον, αναλύοντας έτσι ταυτόχρονα τη ημίτονο και το συνημίτονο της γωνίας όπου πρέπει να ανιχνευθεί (Makrand Farkade). Λόγο του ότι έπρεπε να επανασχεδιαστεί όλο το PCB και να τροποποιηθεί το εξάρτημα όπου στερεώνεται ο hall effect αισθητήρας, δεν έγινε εφαρμογή της παραπάνω μεθόδου ανίχνευσης λόγω έλλειψης χρόνου.

Στο κομμάτι του κώδικα, υπάρχει καθυστέρηση στον PID ελεγκτή εξαιτίας του συγχρονισμού αποστολής δεδομένων με την διαδικασία εφαρμογής του κάθε 25ms. Με την εκτέλεση των δύο αυτών διαδικασιών σε διαφορετικό πυρήνα του ESP32, ίσως εξάλειψε το πρόβλημα αυτό και έτσι ο PID θα έκανε πιο συχνές διορθώσεις μέσα σε ένα δευτερόλεπτο, μειώνοντας έτσι τον χρόνο απόκρισης. Δεν εφαρμόστηκε η μέθοδος διαμοιρασμού των διαδικασιών σε δύο πυρήνες λόγω της πολυπλοκότητάς του και αφού τα αποτελέσματα με τα 25ms καθυστέρησης ήταν ικανοποιητικά. Πολύ πιθανό να υπάρχει και πιο αποτελεσματικός τρόπος αποκωδικοποίησης της πληροφορίας, καθώς όταν το σύστημα λαμβάνει από κάποια εφαρμογή εντολές, παγώνει για μερικά ms.

Στο κομμάτι των εφαρμογών, η mobile δεν παρουσιάζει κάποιο θέμα κατά την λειτουργία της όμως μπορούν να προστεθούν παραπάνω λειτουργίες ώστε να βοηθήσουν την εμπειρία του χρήστη κατά

την χρήση της. Στην εφαρμογή για τον ηλεκτρονικό υπολογιστή, απουσιάζει κάποιο σύστημα αποθήκευσης των παραμέτρων και ρυθμίσεων από χρήση σε χρήση, κάνοντας έτσι απαραίτητη την εισαγωγή των τιμών κάθε φορά μετά από επανεκκίνηση της εφαρμογής. Ερευνήθηκε τρόπος αποθήκευσής τους με χρήση JSON αρχείων στον δίσκο του Η/Υ και ανάγνωση αυτών σε κάθε εκκίνηση, όμως δεν εφαρμόστηκε καθώς δεν θεωρήθηκε απαραίτητο για την λειτουργία του συστήματος και κατηγοριοποιήθηκε ως λειτουργία για την βελτίωση της εμπειρίας χρήστη. Μπορεί να εφαρμοστεί σε μελλοντική επέκταση του συστήματος από τρίτους.

Οι τρόποι με τους οποίους το σύστημα θα μπορούσε να ελεγχθεί θα μπορούσε να διευρυνθεί. Παρόλο που ο PID είναι ένας πολύ καλός ελεγκτής, δεν ταιριάζει πάντα σε όλα τα συστήματα εξαιτίας της δομής του, έτσι με εισαγωγή άλλων ελεγκτών θα είναι δυνατή η εξερεύνηση και εκμάθηση πολλών άλλων ελεγκτών.

Το μέλλον που μπορεί να έχει το παρόν εργαλείο στον χώρο εκμάθησης συστημάτων αυτομάτου ελέγχου είναι λαμπρό, ξεκινώντας από μικρές προσθήκες στο πακέτο. Η πιο σημαντική διαδικασία που μπορεί να γίνει, είναι η πλήρης μαθηματικοποίηση του μοντέλου του συστήματος ώστε αυτό μετά να τοποθετηθεί σε προγράμματα όπως το MATLAB και να γίνει εφαρμογή πολλαπλών ελεγκτών επάνω του χωρίς την παρουσία του πραγματικού συστήματος. Εφόσον θα μπορεί να υπολογιστεί η αντίδραση του συστήματος από τις μαθηματικές συναρτήσεις που το περιγράφουν, μπορούν να δοκιμαστούν πληθώρα ελεγκτών και παραμέτρων χωρίς τον φόβο καταπόνησης του συστήματος από συνεχή χρήση ή εισαγωγή λανθασμένης τιμής. Ένας ακόμη τρόπος όπου μπορεί να ενισχυθεί αυτή η μέθοδος είναι η χρήση του 3D μοντέλου που δημιουργήθηκε με την εισαγωγή του στο MATLAB ή μέσω του Unity, να βελτιωθεί η εμπειρία που θα έχει ο χρήστης παρατηρώντας την αντίδραση του συστήματος σε εικονικό περιβάλλον.

## Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

1528-5588-ND. (χ.χ.). Ανάκτηση από Digikey:

<https://www.digikey.com/en/products/detail/adafruit-industries-llc/5588/16719270>

3D printer Creality CR-10 v2 300. (χ.χ.). Ανάκτηση από 3Dexpert:

<https://www.3dexpert.gr/eshop/el/3d-printer-creality-cr-10-v2-300>

ANDRONICEANU, A., & BURLACU, S. (χ.χ.). INTEGRATION OF EDUCATIONAL TECHNOLOGIES IN UNIVERSITIES AND STUDENTS' PERCEPTION THEREOF.

Block diagram of process control system. (χ.χ.). Ανάκτηση από Polytechnic Hub:

<https://www.polytechnichub.com/block-diagram-process-control-system/>

DELCOLABS. (χ.χ.). DC Servo Motor Controller Using PID Method Trainer. Ανάκτηση από DELCOLABS: <https://www.delcolabs.com/product/dc-servo-motor-controller-using-pid-method-trainer>

Educational technology. (χ.χ.). Ανάκτηση από wikipedia:

[https://en.wikipedia.org/wiki/Educational\\_technology](https://en.wikipedia.org/wiki/Educational_technology)

ESP32-DevKitC V4 . (χ.χ.). Ανάκτηση από Espressif: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/hw-reference/esp32/get-started-devkitc.html>

Espressif. (χ.χ.). ESP32-DevKitC V4. Ανάκτηση από Espressif:

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/hw-reference/esp32/get-started-devkitc.html#get-started-esp32-devkitc-board-front>

Hall Effect. (χ.χ.). Ανάκτηση από hyperphysics: <http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/Hall.html>

Hall Effect. (χ.χ.). Ανάκτηση από byjus: <https://byjus.com/physics/hall-effect/>

Introducing JSON. (χ.χ.). Ανάκτηση από json: <https://www.json.org/json-en.html>

J. Crowe, K. T.-P.-C.-G. (2005). *PID Control New Identification and Design Methods*.

kai-morich. (χ.χ.). *SimpleBluetoothTerminal*. Ανάκτηση από github: <https://github.com/kai-morich/SimpleBluetoothTerminal>

Kuo, B. C. (1987). *Automatic Control Systems*. Prentice Hall PTR, United States.

LINEAR HALL-EFFECT IC. (χ.χ.). Ανάκτηση από diodes:

<https://www.diodes.com/assets/Datasheets/AH49E.pdf>

LM1086 1.5-A. (χ.χ.). Ανάκτηση από Texas Instruments:

[https://www.ti.com/lit/ds/symlink/lm1086.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1637163195810&ref\\_url=https%253A%252F%252Fwww.mouser.de%252F](https://www.ti.com/lit/ds/symlink/lm1086.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1637163195810&ref_url=https%253A%252F%252Fwww.mouser.de%252F)

Makrand Farkade, T. R. (χ.χ.). Hall-effect based sensor application on angular measurement. Ανάκτηση από <https://www.ee.iitb.ac.in/~stallur/wp-content/uploads/2017/02/Angle-measurement-using-Hall-sensor.pdf>

Nagel, L. (χ.χ.). *What is an Electronic Speed Controller & How Does an ESC Work*. Ανάκτηση από tytorobotics: <https://www.tytorobotics.com/blogs/articles/what-is-an-esc-how-does-an-esc-work>

Perens, B. (2007 / 2008). The Open Source Definition.

*PID Control*. (χ.χ.). Ανάκτηση από Circuit Cellar: <https://circuitcellar.com/resources/quickbits/pid-control/>

*Pin Header 1x20 Female 2.54mm*. (χ.χ.). Ανάκτηση από Grobotronics: <https://grobotronics.com/pin-header-1x20-female-2.54mm.html?sl=en>

*Rotary Inverted Pendulum*. (χ.χ.). Ανάκτηση από QUANSER: <https://www.quanser.com/products/rotary-inverted-pendulum/#details>

*Screw Terminal 2P 5mm*. (χ.χ.). Ανάκτηση από Grobotronics: <https://grobotronics.com/screw-terminal-2p-5mm.html?sl=en>

*Speed Control of a DC Motor*. (χ.χ.). Ανάκτηση από WOLFRAM: <https://reference.wolfram.com/language/MicrocontrollerKit/workflow/MotorSpeedControl>

*T-Connector Plug*. (χ.χ.). Ανάκτηση από hellas digital: <https://www.hellasdigital.gr/electronics/connector/dean-connector-t-plug-for-esc-battery/?sl=en>

*What is 3D Printing?* (χ.χ.). Ανάκτηση από 3dprinting: <https://3dprinting.com/what-is-3d-printing/>

*Wires 3-Pin 30cm Female to Female*. (χ.χ.). Ανάκτηση από Grobotronics: <https://grobotronics.com/jumper-wires-3-pin-30cm-female-to-female.html>

*XD A2212 1400KV Brushless Motor*. (χ.χ.). Ανάκτηση από ariesrc: <https://www.ariesrc.gr/en/motors/9510-xxd-a2212-1000kv-brushless-motor-for-rc-airplane-quadcopter.html>

*ΜΑΓΝΗΤΗΣ ΔΙΣΚΟΣ Φ6×2,5mm ΔΙΑΜΕΤΡΙΚΟΣ*. (χ.χ.). Ανάκτηση από Magnitech: <https://magnitech.gr/shop/magnites-neodymiou-isxyroi/magnites-diskoi-neodymiou/magnitis-diskos-f6x25mm-diametrikos-neody/>

Thingiverse. (χ.χ.). *Thingiverse*. Ανάκτηση από Thingiverse: <https://www.thingiverse.com/>

## Παράρτημα Α

Παρακάτω ο κώδικας που γράφηκε για το ESP32:

```
/*
 * This is the main code for the ESP32 MCU of the aeropendulum system
 */

#include <BluetoothSerial.h>
#include <ESP32Servo.h>

// Hardware pin initialization
int hes_pin = 34;
int esc_pin = 32;

// System specific variable initialization
int angle_val;
int pwm_val;
int state;

// PID controller variables
int target_angle;
double error;
double Kp;
double Ki;
double Kd;
double integral = 0;
double derivative;
double prev_angle = 0;
unsigned long now = millis();
unsigned long lastTime;
double elapsedTime;
double dt = 25; //Time for PID in ms

// Library software initialization
BluetoothSerial SerialBT;
Servo ESC;

// Main Setup
void setup() {
  // Pin Modes and Serial Initialization
  pinMode(hes_pin, INPUT);
  Serial.begin(115200);
  SerialBT.begin("Aeropendulum");

  // ESC Initialization
  ESC.setPeriodHertz(50);
  ESC.attach(esc_pin, 1000, 2000);
  // Sets ESC to the default state
  stop();
  delay(3000);
}
```

```
}

// Main Loop
void loop() {
  // Updates angle and handles new serial and BT data
  updateAngle();
  sendData();
  readData();

  // Main logic
  if(state == 1) {
    manual();
  }
  else if(state == 2) {
    PID();
  }
}

// Function of the Manual State
void manual() {
  // Check for extreme angle
  while(angle_val < 85) {
    updateAngle();
    sendData();
    readData();

    if(state == 0){
      stop();
      break;
    }
    ESC.write(pwm_val);
  }
  // Safe motor shut down
  stop();
}

// Function of the PID state
void PID() {
  // Check for extreme angle
  while(angle_val < 85) {
    sendData();
    readData();

    if(state == 0){
      stop();
      break;
    }
  }

  // PID algorithm that executes every dt
  if (elapsedTime >= dt) {
    // Update Angle
```

```
    updateAngle();

    // Calculate PID
    error = target_angle - angle_val;
    integral += (error * elapsedTime);
    derivative = (angle_val - prev_angle) / elapsedTime;

    // Calculate output
    double output = Kp * error + Ki * integral + Kd * derivative;
    output = constrain(output, 0, 100);
    pwm_val = output;

    // Write output to motor
    ESC.write(pwm_val);

    // Update time variable
    prev_angle = angle_val;
}
}
// Resets integral part
integral = 0;
// Safe motor shut down
stop();
}

// Reads encoded data and decodes it from both serial and BT
void readData(){
    String txt = "";
    if (SerialBT.available()) {
        txt = SerialBT.readString();
        txt.trim();
        decodeData(txt);
    }
    else if (Serial.available()) {
        txt = Serial.readString();
        txt.trim();
        decodeData(txt);
    }
}

// Encodes and sends all data
void sendData() {
    now = millis();
    elapsedTime = (double)(now - lastTime);

    // Sends data every dt
    if (elapsedTime >= dt) {
        // Reset timer
        lastTime = now;

        // Formats data
```



```
String angleF = formatData(angle_val);
String pwm_valF = formatData(pwm_val);

// Encodes the data to be sent
String dataStream = "A" + angleF + "P" + pwm_valF + "S" + state;
sendT(dataStream);
}
}

// Formats data to add extra "0" at the front
String formatData (int dat) {
  if (dat < 0) {
    return String("-") + (dat >= -9 ? "0" : "") + abs(dat);
  } else {
    if (dat == 100)
    {
      return String(dat);
    }
    else
    {
      return String("0") + (dat <= 9 ? "0" : "") + String(dat);
    }
  }
}

// Decodes the input data and updates all status variables
void decodeData(String data) {
  // Position of data
  int pS = 1, pV = 3, pP = 7, pI = 18, pD = 29, pA = 40;

  // Extracts numbers from the string
  state = data.substring(pS).toInt();
  pwm_val = data.substring(pV, pV + 3).toInt();
  Kp = data.substring(pP, pP + 10).toDouble();
  Ki = data.substring(pI, pI + 10).toDouble();
  Kd = data.substring(pD, pD + 10).toDouble();
  target_angle = data.substring(pA, pA + 3).toInt();
}

// Reads sensor and returns angle data
void updateAngle() {
  angle_val = map((analogRead(hes_pin) - 955),0,1755,-90,90);
}

// Sends any string in a new line in both serial and BT
void sendT(String str) {
  SerialBT.println(str);
  Serial.println(str);
}

// Stops the motor from moving and changes all appropriate parameters
```

```
void stop() {  
    pwm_val = 0;  
    ESC.write(pwm_val);  
    state = 0;  
}
```