ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών
«ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ ΚΑΙ ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδίαση, Ανάπτυξη και Προγραμματιζόμενος Έλεγχος μη επανδρωμένου σκάφους επιφανείας με τη χρήση τεχνολογιών του διαδικτύου των πράγματων»

Μεταπτυχιακός Φοιτητής: Θεόδωρος Δρυμώνης, ΑΜ: msciot20004
Επιβλέπων: Διονύσης Κανδρής, Καθηγητής

ΑΙΓΑΛΕΩ, ΙΟΥΝΙΟΣ 2022

UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Master of Science in

"INTERNET of THINGS AND INTELLIGENT ENVIRONMENTS"

MSc Thesis

"Design, Development and Programmable Control of an Unmanned Surface Vessel using Internet of Things technologies"

Student: Theodoros Drymonis, Registration Number: msciot20004

MSc Thesis Supervisor: Dionisis Kandris, Professor

ATHENS-EGALEO, JUNE 2022

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

Διονύσης Κανδρής, Καθηγητής	Γρηγόρης Καλτσάς, Καθηγητής	Παναγιώτης Παπαγέωργας, Καθηγητής
Dionisis Kandris Digitally signed by Dionisis Kandris Date: 2022.06.28 23:55:53 +03'00'	Digitally signed by Grigorios Kaltsas Date: 2022.06.29 09:42:08 +03'00'	Digitally signed by Panagiotis Papageorgas Date: 2022.06.29 09:52:03 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

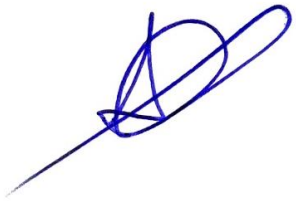
Ο κάτωθι υπογεγραμμένος Θεόδωρος Δρυμώνης του Παναγιώτη, με αριθμό μητρώου msciot20004 μεταπτυχιακός φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ, στο ΠΜΣ «Διαδίκτυο των Πραγμάτων και Ευφυή Περιβάλλοντα»

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.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο Δηλών



Θεόδωρος Δρυμώνης

ΠΕΡΙΛΗΨΗ

Στην εποχή μας, παρατηρείται η ενσωμάτωση ολοένα και περισσοτέρων μηχανών στον κόσμο του διαδικτύου των πραγμάτων (IoT, Internet of Things). Σκοπός της διατριβής αυτής είναι να διερευνηθούν οι δυνατότητες ενσωμάτωσης των IoT τεχνολογιών για τον έλεγχο ενός αυτόνομου ρομποτικού συστήματος. Πιο συγκεκριμένα, το σύστημα πάνω στο οποίο έγινε η διερεύνηση, είναι ένα μη επανδρωμένο σκάφος επιφανείας, το οποίο σχεδιάστηκε, αναπτύχθηκε και δοκιμάστηκε η λειτουργία του σε πραγματικές συνθήκες.

Ο έλεγχος του οχήματος, βασίστηκε σε τεχνολογίες του διαδικτύου των πραγμάτων. Για την επικοινωνία μεταξύ μηχανών (MTC) και την μετάδοση δεδομένων στο διαδίκτυο, χρησιμοποιήθηκε το πρωτόκολλο MQTT. Η διαχείριση του οχήματος και των πληροφοριών που σχετίζονται με αυτό, έγινε μέσω της πλατφόρμας υπηρεσιών Node-RED (IoT PaaS). Για να επιτευχθεί ο απομακρυσμένος έλεγχος η παραπάνω υπηρεσία και ο MQTT Server/Broker αναπτύχθηκαν σε ένα υπολογιστικό σύστημα νέφους (Cloud Computer). Η οδήγηση του ρομποτικού σκάφους βασίστηκε στην δημοφιλή, ανοικτού κώδικα πλατφόρμα αυτόματου πιλότου Ardupilot και έγινε εκτεταμένη χρήση του αντίστοιχου πρωτόκολλου επικοινωνίας MAVLink. Απαραίτητη ήταν η δημιουργία μιας εφαρμογής σε πλατφόρμα ανάπτυξης μικροελεγκτή ESP32, ώστε να γεφυρωθούν τα πρωτόκολλα MQTT και MAVLink.

Στο πλαίσιο της εργασίας αυτής, γίνεται αναφορά στις σχετικές τεχνολογίες και βασικές θεωρητικές αρχές. Έπειτα παρουσιάζεται η μελέτη σχετικά με την αρχιτεκτονική του συστήματος και το υλικό, λογισμικό που επιλέχθηκαν να χρησιμοποιηθούν. Επίσης περιγράφεται, η διαδικασία υλοποίησης του ρομποτικού οχήματος, ανά υποσύστημα και όπου χρειάστηκε αναφέρθηκαν τα προβλήματα που προέκυψαν, αλλά και πώς αντιμετωπίστηκαν. Επιπρόσθετα παρουσιάζονται, τα αποτελέσματα και τα συμπεράσματα που προέκυψαν από τις δοκιμές του οχήματος. Τέλος προτείνονται μελλοντικά αντικείμενα έρευνας για τη βελτίωση των επιμέρους υποσυστημάτων του μη επανδρωμένου σκάφους επιφανείας που αναπτύχθηκε.

ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ: Μη επανδρωμένο σκάφος επιφανείας, διαδίκτυο των πραγμάτων, ρομπότ, ελεγκτής πλοήγησης, Ardupilot, MAVLink, MQTT, Node-RED, νεφοϋπολογιστική, μικροελεγκτής.

ABSTRACT

Nowadays, more and more machines are getting integrated into the world of the Internet of Things (IoT, Internet of Things). The purpose of the present thesis is to explore the possibilities of integrating IoT technologies to control an autonomous robotic system. More specifically, the research was carried out on an unmanned surface vessel, which was designed, developed and tested in real conditions.

The control of this vessel was based on the use of technologies related with the Internet of Things. Specifically, the MQTT protocol was used for machine type communication (MTC) and data transfer over the internet. Also, the vehicle as well as the information related to it were managed through the Node-RED service platform (IoT PaaS). In order to achieve remote control, the aforementioned services together with the MQTT Server/Broker were developed on cloud computing system (Cloud Computer). Additionally, the robotic vehicle was based on the popular, open-source coding platform for automatic pilot Ardupilot and extensive use of the corresponding communication protocol MAVLink was made. Moreover, an application on the ESP32 microcontroller development platform was createad in order to bridge the MQTT and MAVLink protocols.

In the context of this work, the reader is introduced to the relevant technologies and their basic theoretical principles. Next, the study on the system architecture, and the chosen hardware and software are presented. Also, the implementation process of the robotic vehicle per subsystem is described. Additionally, problems that occurred are mentioned along with the procedures that were adopted in order to address these problems. Furthermore, the results of the vehicle tests along with conclusions drawn are presented. Finally, future research is proposed regarding the improvement of the individual subsystems of the unmanned surface vessel that was developed.

KEYWORDS: Unmanned surface vessel, internet of things, robot, flight controller, Ardupilot, MAVLink, MQTT, Node-RED, cloud computing, microcontroller.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Διονύση Κανδρή για την καθοδήγηση που μου έδωσε κατά τη διάρκεια εκπόνησης της διπλωματικής μου, καθώς επίσης και για τις γνώσεις που μου προσέφερε σε όλη τη διάρκεια των σπουδών μου.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τη σύντροφο μου για την υποστήριξη που μου προσέφεραν.

Τέλος, θέλω να ευχαριστήσω τους φίλους και τους συνάδελφούς μου από την Ελληνική Αεροπορική Βιομηχανία, που και αυτοί στήριξαν την προσπάθεια μου και βοήθησαν με τον τρόπο τους στην επιτυχή ολοκλήρωση αυτού του δύσκολου έργου.

ΚΑΤΑΛΟΓΟΣ ΑΚΡΩΝΥΜΙΩΝ - ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

4G: Fourth Generation (mobile communications network)

5G: Fifth Generation (mobile communications network)

AHRS: Attitude and Heading Reference System

AP: Access point

BEC: Battery Eliminator Circuit (DC to DC Converter or Voltage Regulator)

CAN: Communication Area Network (Device Communication Bus)

CRC: Cyclic Redundancy Check (Algorithm)

ESC: Electronic Speed Control

GCS: Ground Control Station

GNSS: Global Navigation Satellite System

I2C: Inter Integrated Circuit (Embedded systems serial communication bus)

IMU: Inertia Measurement Unit

IoT: Internet of Things

IP: Internet Protocol

ISM: Industrial Scientific Medical (Radio frequency bands)

IEEE: The Institute for Electrical and Electronics Engineers

LDO: Low Dropout (Linear Voltage Regulator)

LiPo: Lithium Polymer (Battery Technology)

LPWAN: Low Power Wide Area Network

M2M: Machine to Machine

MAVLink: Micro Air Vehicle Link (Unmanned Vehicle Communication Protocol)

MCU: Microcontroller Unit

MQTT: Message Queue Telemetry Transport

MTC: Machine Type Communications

NB-IoT: NarrowBand Internet of Things

OSD: On Screen Display

PaaS: Platform as a Service

PDB: Power Distributor Board

PWM: Pulse Width Modulation

QoS: Quality of Service

RTL: Return to Launch

SPI: Serial Peripheral Interface (Embedded systems serial communication bus)

UART: Universal Asynchronous Receiver/Transmitter

UDP: User Datagram Protocol

VTOL: Vertical Takeoff And Landing Airplane

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΕΙΣΑΓΩΓΗ

Η πρόοδος των επιστημών και της τεχνολογίας σε όλους του τομείς έχει εισαγάγει στη ζωή των ανθρώπων καινοτόμα προϊόντα που έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας μας. Οι τεχνολογίες της μικροηλεκτρομηχανικής και της λιθογραφίας έχουν επιτρέψει την ανάπτυξη αισθητήρων μικροεπεξεργαστών και άλλων λογικών κυκλωμάτων σε εξαιρετικά μικρές κλίμακες. Ως εκ τούτου είναι πλέον δυνατή η δημιουργία προηγμένων ενσωματωμένων συστημάτων που με τη σειρά τους έχουν φέρει επανάσταση στον χώρο της μηχατρονικής. Ρομπότ που στο παρελθόν, ήταν στη σφαίρα της φαντασίας μας ή στην καλύτερη περίπτωση, ήταν σε πειραματικό στάδιο και πολύ ογκώδη για να έχουν πρακτική χρήση, πλέον κατακλύζουν την αγορά και απευθύνονται σε ολοένα και περισσότερες εφαρμογές [1] [2].

Εικόνα 0-1. Πρωτότυπο Drone με ενσωματωμένο ρομποτικό βραχίονα [1].

Εικόνα 0-2. RoMan (Robotic Manipulator) [2].

Η αξιοπιστία και η αποτελεσματικότητα των συστημάτων αυτών, σε συνδυασμό με την ραγδαία εξέλιξη της τεχνητής νοημοσύνης, έχει οδηγήσει στην αυτοματοποίηση όλων των οχημάτων που για την οδήγηση τους προηγουμένως, ο ανθρώπινος παράγοντας ήταν απαραίτητος.

Επιπρόσθετα καινοτομίες στο τομέα της πληροφορικής και του διαδικτύου εισήγαγαν την ιδέα του διαδικτύου τον πραγμάτων (IoT, Internet of Things). Πιο συγκεκριμένα η ιδέα αυτή βασίζεται στην διασύνδεση των φυσικών αντικειμένων με τον ψηφιακό κόσμο [3] [4]. Τα ηλεκτρονικά συστήματα, οι ενεργοποιητές και οι αισθητήρες, έχουν πλέον έχουν δυνατότητα να επικοινωνούν μέσω ενός τοπικού δικτύου ή ακόμα και μέσω του παγκόσμιου ιστού Internet και χρησιμοποιούν τεχνολογίες όπως:

- τα πρωτόκολλα επικοινωνίας M2M (machine to machine) που βασίζονται στο υπάρχον πρωτόκολλο των δικτύων υπολογιστών (TCP/IP),
- νεφοϋπολογιστική
- και ασύρματες επικοινωνίες όπως το 4G και το 5G [5].

Το διαδίκτυο των πραγμάτων έχει γίνει κομμάτι της καθημερινότητας μας τη τελευταία δεκαετία, ο αριθμός των συσκευών που έχουν IoT δυνατότητες αυξάνεται κατακόρυφα. Έξυπνες οικιακές συσκευές που ελέγχονται μέσω smartphone και η αυτόματη διαχείριση ενέργειας στα έξυπνα σπίτια είναι μερικές από τις εφαρμογές του διαδικτύου των πραγμάτων [6].

Είναι προφανές ότι η ενσωμάτωση των ρομπότ και των μη επανδρωμένων οχημάτων στον κόσμο των IoT, είναι το επόμενο στάδιο για τη βελτίωση του βιοτικού επιπέδου, αλλά και τη δημιουργία νέων προϊόντων και υπηρεσιών. Ως εκ τούτου, μεγάλοι βιομηχανικοί κολοσσοί όπως η Amazon, Volvo, Tesla και General Motors επενδύουν δισεκατομμύρια δολάρια στην έρευνα για την δημιουργία αυτόνομων οχημάτων. Ήδη η τεχνολογία τους θεωρείται αρκετά ώριμη και οι πρώτοι στόλοι αυτόνομων οχημάτων κάνουν την εμφάνιση τους. Η υπηρεσία ταχυμεταφορών με drone της Amazon, γνωστή και ως “amazon prime air” και τα αυτόνομα μέσα μαζικής μεταφοράς της General Motors είναι από τις πρώτες μορφές υπηρεσιών που συνδυάζουν την τεχνολογία του διαδικτύου των πραγμάτων και των μη επανδρωμένων οχημάτων [7] [8].

Αντικείμενο της διπλωματικής εργασίας

Στην εργασία αυτή πραγματοποιήθηκε η σχεδίαση και ανάπτυξη ενός μη επανδρωμένου σκάφους επιφανείας, το οποίο μπορεί να ελεγχθεί μέσω μίας διαδικτυακής πλατφόρμας υπηρεσιών. Το αντικείμενο αυτό έχει ενδιαφέρον διότι μέχρι τώρα, η πλειονότητα, τέτοιων ρομποτικών οχημάτων χρειάζεται να συνοδεύονται το κάθε ένα χωριστά από εξειδικευμένο εξοπλισμό και εκπαιδευμένο προσωπικό για την χρήση και την επιτήρηση κατά τη διάρκεια λειτουργίας τους. Επομένως είναι ένα σημαντικό θέμα διερεύνησης, καθώς η ενσωμάτωση των IoT τεχνολογιών στα μη επανδρωμένα οχήματα δίνει τη δυνατότητα να συγκεντρωθεί και να συντονιστεί ο έλεγχος μεγάλου όγκου αυτόνομων οχημάτων. Επιπλέον η χρήση των τεχνολογιών νέφους, της τεχνητή νοημοσύνης και των ταχύτατα αναπτυσσομένων υποδομών τηλεπικοινωνιών, μπορεί να επεκτείνει σημαντικά την εμβέλεια κάλυψης αλλά και το εύρος εφαρμογών που μπορούν να χρησιμοποιηθούν.

Σκοπός και στόχοι

Σκοπός της παρούσας διπλωματικής εργασίας είναι να χρησιμοποιηθεί η μεθοδολογία σχεδιασμού και ανάπτυξης του ρομποτικού οχήματος ώστε να αποκτήσει αυτόνομη λειτουργία σε πραγματικές συνθήκες, με την ενσωμάτωση τεχνολογιών του διαδικτύου των πραγμάτων. Στόχος είναι η απομακρυσμένη διαχείριση και επιτήρηση της αποστολής του οχήματος χρησιμοποιώντας πλατφόρμες που βασίζονται στην νεφοϋπολογιστική. Όπως επίσης να παρουσιαστεί ένας τρόπος διασύνδεσης δύο πρωτοκόλλων που συνηθίζεται να συνεργάζονται σε εφαρμογές ρομποτικής.

Μεθοδολογία

Το έργο που εκπονήθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας χωρίστηκε σε τέσσερα βασικά στάδια.

Στο πρώτο στάδιο, διερευνήθηκαν οι όλες οι τεχνολογίες που σχετίζονται με το αντικείμενο της διπλωματικής, και μελετήθηκε το θεωρητικό υπόβαθρο και οι εφαρμογές τους. Πιο συγκεκριμένα, πραγματοποιήθηκε εκτεταμένη αναζήτηση σχετικών επιστημονικών δημοσιεύσεων και άλλων πηγών του διαδικτύου. Αφού ολοκληρώθηκε η βιβλιογραφική και διαδικτυακή έρευνα με γνώμονα τα ρομποτικά οχήματα και τις διάφορες τεχνολογίες που ενσωματώνουν, παρατηρήθηκε ότι δεν υπάρχουν αρκετές υλοποιήσεις μη επανδρωμένων σκαφών επιφανείας που να συνδυάζουν τεχνολογίες του διαδικτύου των πραγμάτων.

Σε δεύτερο στάδιο έγινε αναζήτηση παρεμφερών υλοποιήσεων από τις κοινότητες ανοιχτού υλικού και λογισμικού, καθώς και του εμπορίου. Ως εκ τούτου έγινε προσπάθεια να καταγραφούν οι αρχιτεκτονικές πάνω στις οποίες βασίζονται αυτά τα συστήματα και να βγουν συμπεράσματα για τα πλεονεκτήματα και μειονεκτήματα των τεχνολογιών που η κάθε υλοποίηση ρομποτικού οχήματος ενσωματώνει. Με γνώμονα τις πληροφορίες που συγκεντρώθηκαν, σχεδιάστηκε η αρχιτεκτονική του μη επανδρωμένου οχήματος της παρούσας διπλωματικής και στη συνέχεια έγινε έρευνα αγοράς των υλικών που χρειάστηκαν για τη κατασκευή.

Στο τρίτο στάδιο της εργασίας, έγινε η σταδιακή περισυλλογή και η συναρμολόγηση των υλικών, ξεκινώντας από τα απολύτως βασικά και συνεχίζοντας στα πιο ανεπτυγμένα υποσυστήματα. Βασικής σημασίας, ήταν η απόκτηση ενός ευρύχωρου υδροδυναμικού μοντέλου ταχύπλοου σκάφους, το οποίο ελέγχθηκε για την πλευστότητα και για τυχόν διαρροές υδάτων, ενώ όπου χρειάστηκε έγιναν οι απαραίτητες επισκευές. Στη συνέχεια κατασκευάστηκε η εσωτερική δομή του σκάφους ώστε να υπάρχουν οι απαραίτητες θέσεις για τον ηλεκτρομηχανικό εξοπλισμό και τα λοιπά ηλεκτρονικά συστήματα που επρόκειτο να τοποθετηθούν. Όπως αναφέρθηκε προηγουμένως, τα απολύτως απαραίτητα ηλεκτρονικά και ηλεκτρομηχανικά εξαρτήματα όπως, η μπαταρία, το σύστημα τηλεκατεύθυνσης, ο κινητήρας με το ηλεκτρονικό του κύκλωμα ελέγχου και τέλος ο σερβομηχανισμός ελέγχου του πηδαλίου τοποθετήθηκαν πρώτα. Το σκάφος εξοπλισμένο πλέον με τα βασικά εξαρτήματα δοκιμάστηκε εκτενώς, άμεσα ελεγχόμενο από το χρήστη μέσω τηλεκατεύθυνσης, με σκοπό να γίνει έλεγχος καταπόνησης όλων των υποσυστημάτων αλλά και να δοκιμαστεί η σταθερότητα του κατά την οδήγηση. Ένας κύκλος δοκιμών και

διορθώσεων πραγματοποιήθηκε, έως ότου η βασική υλοποίηση του σκάφους είχε προβλέψιμα και αξιόπιστα χαρακτηριστικά.

Στο τέταρτο και τελευταίο μέρος, τα πιο ανεπτυγμένα ηλεκτρονικά στα οποία βασίζεται το κυρίως αντικείμενο της εργασίας αυτής, τοποθετήθηκαν μέσα από ένα κύκλο δοκιμών και διορθώσεων. Αναπτύχθηκε σταδιακά το λογισμικό όλων των υποσυστημάτων και έγιναν οι απαραίτητες αναπροσαρμογές μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα. Στη φάση αυτή καταγράφηκαν όλα τα συμπεράσματα, οι παρατηρήσεις και οι προτάσεις για μελλοντική βελτίωση του μη επανδρωμένου σκάφους επιφανείας.

Καινοτομία

Η καινοτομία της διατριβής έγκειται στη συνδυασμένη χρήση διαθέσιμων και χαμηλού κόστους, τεχνολογιών και υλικών, για τη δημιουργία μιας αρχιτεκτονικής ρομποτικού οχήματος, που συνδυάζει πρωτόκολλα επικοινωνίας M2M, υπηρεσιών νεφοϋπολογιστικής και πρωτόκολλων επικοινωνίας μη επανδρωμένων οχημάτων. Επομένως το σύστημα που σχεδιάστηκε και υλοποιήθηκε, χάρη στην απομακρυσμένη διαχείριση του μέσω υπηρεσιών νεφοϋπολογιστικής, μπορεί εύκολα να επεκταθεί και να προσαρμοστεί σε διάφορους τύπους αποστολών.

Δομή

Στο πρώτο κεφάλαιο, γίνεται μια εισαγωγή στα ρομποτικά οχήματα. Δίνεται μια γενική περιγραφή της αρχιτεκτονικής των οχημάτων επιφανείας και της δομής των συστημάτων καθοδήγησης, πλοήγησης και ελέγχου.

Στο δεύτερο κεφάλαιο, παρουσιάζεται το θεωρητικό υπόβαθρο των τεχνολογιών που θα χρησιμοποιηθούν στο μη επανδρωμένο σκάφος της παρούσας υλοποίησης.

Στο τρίτο κεφάλαιο περιγράφεται η διαδικασία υλοποίησης του σκάφους χωρίζοντας τη σε δύο μέρη. Το πρώτο μέρος περιλαμβάνει τη κατασκευή των μηχανικών και ηλεκτρονικών συστημάτων. Το δεύτερο μέρος περιλαμβάνει το προγραμματισμό του λογισμικού που χρησιμοποιήθηκε.

Στο τέταρτο κεφάλαιο, γίνεται η σύνοψή της εργασίας, αναλύονται τα προβλήματα που εμφανίστηκαν και ο τρόπος αντιμετώπισης τους. Τέλος γίνεται αξιολόγηση του τελικού αποτελέσματος και προτείνονται αντικείμενα μελλοντικής εξέλιξης σχετικά με το θέμα της εργασίας.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στη Ρομποτική και τα ρομποτικά οχήματα

Στο πέρασμα του χρόνου υπήρξε μεγάλη εξέλιξη στο τομέα της τεχνολογίας. Από την ανακάλυψη του τροχού μέχρι σήμερα έχουν αναπτυχθεί νέοι μηχανισμοί. Ο τομέας της ηλεκτρονικής με τους ημιαγωγούς και της πληροφορικής με τη χρήση του προγραμματισμού, έχουν βοηθήσει στην ραγδαία ανάπτυξη των αυτοματισμών. Η αυξημένη χρήση των αυτοματισμών και η ενσωμάτωση τους στην βιομηχανία και την καθημερινότητα έχουν αρχίσει να αντικαθιστούν τον ανθρώπινο παράγοντα με ρομπότ. Στο κεφάλαιο αυτό θα γίνει αναφορά στην εξέλιξη των ρομπότ και των ρομποτικών οχημάτων. Επίσης θα γίνει μία σύντομη αναφορά στα είδη των ρομποτικών οχημάτων επιφανείας και κάποιων κοινών χαρακτηριστικών τους.

1.1 Ρομποτική και ρομποτικά οχήματα

Η ρομποτική είναι ο τεχνολογικός τομέας που ασχολείται με την σχεδίαση και ανάπτυξη ρομποτικών διατάξεων. Οι ρομποτικές διατάξεις μπορούν να είναι συσκευές ή οχήματα με διαφορετική δομή και τρόπο λειτουργίας και μπορεί να εξυπηρετούν σκοπούς πολλών τομέων. Για τον λόγο αυτό, η ανάπτυξη τους απαιτεί από τον μηχανικό να έχει ένα ευρύ υπόβαθρο γνώσεων που περιλαμβάνει μηχανολογικές , ηλεκτρονικές/ηλεκτρολογικές και προγραμματιστικές γνώσεις.

Το Ρομπότ θα μπορούσε να περιγραφεί ως ένα υλικό μέσο που είναι ικανό να εκτελεί κίνηση για την επίτευξη εργασιών. Ο βαθμός αυτονομίας ενός ρομπότ εξαρτάται από την ικανότητα να εκτελεί μια διατεταγμένη σειρά αντίληψης, λήψης αποφάσεων και δράσης [9].

Αυτό που διακρίνει ένα ρομπότ από ένα ρομποτικό όχημα είναι η ικανότητα του να κινείται στον χώρο. Το ρομποτικό όχημα είναι δηλαδή ένα ρομπότ το οποίο δεν περιέχει οδηγό και ο χειριστής εξ' αποστάσεως αναλαμβάνει τον έλεγχο του οχήματος όπου απαιτείται. Το ρομποτικό όχημα δύναται να περιλαμβάνει και αυτόματο πιλότο, που είναι μια συσκευή η οποία επιτρέπει στο ρομποτικό όχημα να κινείται αυτόνομα στο χώρο. Φυσικά, ο χειριστής μπορεί να έχει την επιλογή να απενεργοποιήσει οποιαδήποτε στιγμή τον αυτόματο πιλότο και να αποκτά τον πλήρη έλεγχο του οχήματος χειροκίνητα [10].

1.2 Κατηγορίες ρομποτικών οχημάτων

Τα ρομποτικά οχήματα διαφοροποιούνται με βάση το χώρο στον οποίο κινούνται και είναι τα εξής:

- **Εναέρια οχήματα**: αποτελούνται από τα αεροπλάνα σταθερής πτέρυγας ή VTOL και τα ελικοφόρα με ένα ή περισσότερούς ρότορες. Ελέγχονται αποκλειστικά από ασύρματα συστήματα τηλεπικοινωνιών. Ειδικεύονται σε στρατιωτικές εφαρμογές, στην παρακολούθηση και καταγραφή δεδομένων σχετικά με την επιφάνεια του εδάφους και των ατμοσφαιρικών συνθηκών.

Εικόνα 1-1. Εναέριο ρομποτικό όχημα σταθερής πτέρυγας [11].

- **Οχήματα υποβρύχια και επιφανείας**: τα οχήματα επιφάνειας επιπλέουν στο νερό και συνήθως ελέγχονται ασύρματα όπως και τα εναέρια οχήματα. Στα υποβρύχια οχήματα είναι πιο περιορισμένος ο έλεγχος, διότι το νερό εμποδίζει τη μετάδοση των ραδιοκυμάτων και γι' αυτό χρησιμοποιούνται ειδικά καλώδια επικοινωνίας που αντέχουν στο νερό [10].

Εικόνα 1-2. Ρομποτικό όχημα επιφανείας [12].

- **Οχήματα εδάφους**: είναι τα οχήματα που κινούνται στο έδαφος, γι' αυτό διαθέτουν διαφορετικά εξαρτήματα και μηχανισμούς για την ώθηση τους, αναλόγως τον τύπο της επιφάνειας του εδάφους που θα κινηθούν. Επομένως τα οχήματα εδάφους χωρίζονται κυρίως σε τροχοφόρα με διαφορετικό τύπο ελαστικών και αριθμό τροχών και τα ερπιστιοφόρα [10].

Εικόνα 1-3. Ρομποτικό όχημα εδάφους [13].

1.3 Ιστορική αναδρομή για τα ρομποτικά οχήματα επιφανείας

Το 1993 αναπτύχθηκε το πρώτο ρομποτικό όχημα επιφάνειας για διάφορες αποστολές από το πρόγραμμα MIT Sea Grant College και ονομάστηκε ARTEMIS. Αυτό το σκάφος ήταν ένα πιστό αντίγραφο αλιευτικής μηχανότρατας και χρησιμοποιήθηκε για τη δοκιμή συστημάτων πλοήγησης και ελέγχου και στη συνέχεια για τη συλλογή δεδομένων βαθυμετρίας στο ποταμό Charles στη Βοστώνη. Επειδή, το σκάφος ARTEMIS ήταν μικρό σε μέγεθος, είχε ως αποτέλεσμα το περιορισμό στην αντοχή του και τη ναυσιπλοΐα του, γι' αυτό οι προδιαγραφές του επόμενου σκάφους βασίστηκαν στην επιθυμία να δημιουργηθεί ένα σύστημα τόσο ευέλικτο και χρήσιμο όσο ένα μικρό σκάφος για να είναι εύκολο για εργασίες ανάπτυξης και έρευνας. Το 1996 αναπτύχθηκε το καινούργιο σκάφος που ονομάστηκε ACES και το 1997 εξοπλίστηκε με αισθητήρες για υδρογραφική έρευνα ολοκληρώνοντας με επιτυχία τη πρώτη δοκιμή στο λιμάνι της Βοστόνης το 1997. Στη συνέχεια το 1998 έγιναν μηχανολογικές αναβαθμίσεις στο σκάφος μέχρι το 2000 που ονομάστηκε AutoCat, όπως φαίνεται στην Εικόνα 1-4 [14].

Εικόνα 1-4. Όχημα επιφανείας AutoCat [14].

Αυτές οι πρώτες έρευνες ήταν η έμπνευση για να ξεκινήσουν και άλλα προγράμματα για την ανάπτυξη ρομποτικών οχημάτων επιφανείας για διάφορες αποστολές. Το πολεμικό ναυτικό της Αμερικής, ανέπτυξε σκάφη επιφανείας για τη ασφάλεια του λιμανιού και για την σάρωση ναρκών. Μέχρι και τώρα, οι εφαρμογές έρευνας έχουν κάνει χρήση μια ποικιλίας μορφών από σκάφη επιφανείας για την επίτευξη διάφορων αποστολών [14].

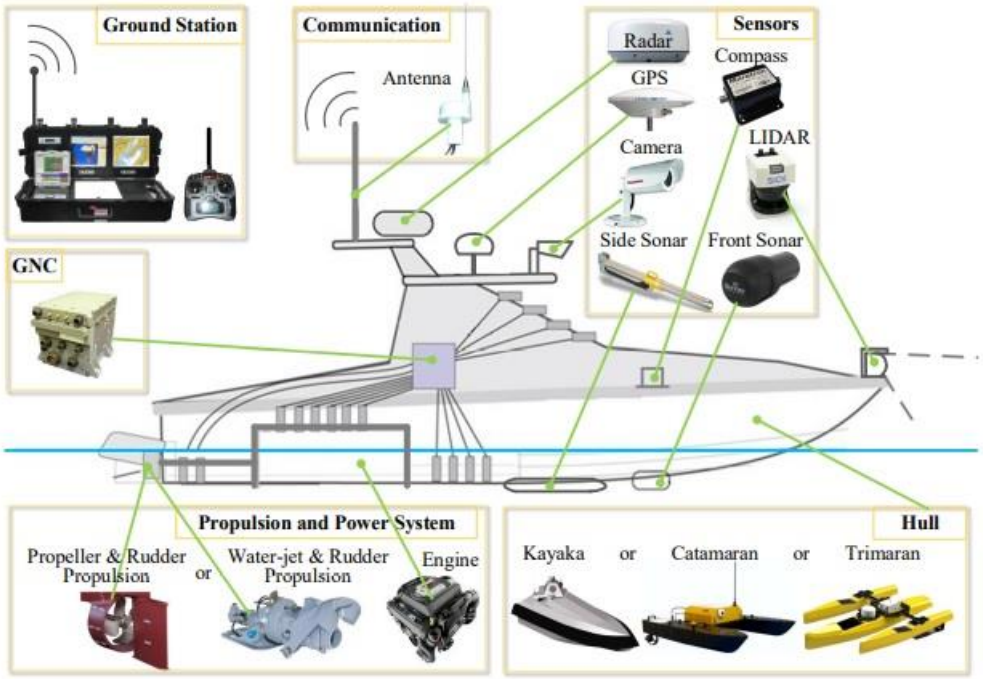
1.4 Η αρχιτεκτονική ενός τυπικού ρομποτικού οχήματος επιφανείας

Ανάλογα με τις εφαρμογές, υπάρχει ποικιλία στα σκάφη και στις λειτουργίες τους. Παρ' όλα αυτά, τα βασικά στοιχεία της αρχιτεκτονικής που πρέπει να περιλαμβάνουν τα ρομποτικά οχήματα επιφανείας είναι τα εξής:

1. **Σκελετός πλοίου και βασικά δομικά στοιχεία (Hull)**: μπορούν να ομαδοποιηθούν σε 4 είδη που είναι το φουσκωτό, καγιάκ, καταμαράν και τριμαράν. Αυτά τα τέσσερα είδη, αντιστοιχούν σε διαφορετικές εφαρμογές αποκαλύπτοντας ορισμένα βασικά ζητήματα σχεδιασμού και τάσεις ανάπτυξης σκαφών επιφανείας. Για παράδειγμα, τα φουσκωτά είναι κατάλληλα για στρατιωτικές εφαρμογές λόγω της μεγάλης αντοχής και της χωρητικότητας τους σε ωφέλιμο φορτίο, ενώ τα καγιάκ και τα καταμαράν παρέχουν ευκολία στην τροποποίηση τους. Επίσης, τα καταμαράν και τριμαράν ξεχωρίζουν λόγω της μεγαλύτερης σταθερότητας που προσφέρουν απέναντι στις καιρικές συνθήκες καθώς και για την παροχή μεγάλης χωρητικότητας σε ωφέλιμο φορτίο [15].
2. **Πρόωση και σύστημα ισχύος (Propulsion and Power System)**: ο έλεγχος κατεύθυνσης και ταχύτητας των περισσότερων υπαρχόντων σκαφών παρέχονται από συστήματα πρόωσης (κινητήρα), πηδαλίου και προπέλα αντίστοιχα. Άλλα σκάφη οδηγούνται από διαφορετικές τεχνικές ώθησης, που παρέχεται από δύο ανεξάρτητους κινητήρες προσαρτημένοι σε κάθε κομμάτι του σκελετού του πλοίου [15]. Κάθε μηχανισμός κατεύθυνσης και προώθησης δίνει διαφορετικές ιδιότητες στον τρόπο που κίνησης του οχήματος, ανάλογα την εφαρμογή για την οποία σχεδιάστηκε.
3. **Συστήματα Guidance Navigation Control (GNC)** : είναι το πιο ζωτικό στοιχείο ενός σκάφους επιφανείας, οι μονάδες **GNC** αποτελούνται γενικά από ενσωματωμένους υπολογιστές και λογισμικό, τα οποία μαζί είναι υπεύθυνα για τη διαχείριση ολόκληρου του συστήματος του σκάφους [15].
4. **Συστήματα επικοινωνίας (Communication)**: τα συστήματα επικοινωνίας περιλαμβάνουν τις ηλεκτρονικές διατάξεις (πομποδέκτες) και τα αντίστοιχα πρωτόκολλα (διαμόρφωση, κωδικοποίηση) για την ασύρματη επικοινωνία με σταθμούς ελέγχου που βρίσκονται στο έδαφος και με άλλα οχήματα για την εκτέλεση συνεργατικού ελέγχου, αλλά και την ενσύρματη ή ασύρματη επικοινωνία με διάφορους αισθητήρες, ενεργοποιητές και άλλο εξοπλισμό. Η αξιοπιστία των συστημάτων επικοινωνίας είναι επομένως υψίστης σημασίας [15].
5. **Εξοπλισμός συλλογής δεδομένων (Sensors) :** οι κάμερες, το ραντάρ, το σόναρ, καθώς και άλλα είδη αισθητήρων τοποθετούνται ανάλογα με την συγκεκριμένη εργασία. Τέτοιες εφαρμογές μπορεί να σχετίζονται με τη παρακολούθηση του περιβάλλοντος, όπως η βυθομέτρηση και η καταγραφή της χλωρίδας και πανίδας. Άλλες εφαρμογές έχουν να κάνουν με τα υποσυστήματα του οχήματος, σε διαφορετικές συνθήκες

περιβάλλοντος όπως η θερμοκρασία κινητήρα, η υγρασία στον χώρο των ηλεκτρονικών συστημάτων και η κατανάλωση καυσίμου [15].

6. **Σταθμός εδάφους (Ground Station)**: ο επίγειος σταθμός παίζει επίσης σημαντικό ρόλο διότι, μπορεί να βρίσκεται σε χερσαία εγκατάσταση, κινητό όχημα ή πλοίο που βρίσκεται στα ανοικτά της θάλασσας. Γενικά, οι αποστολές ανατίθενται στα σκάφη μέσω συστημάτων ασύρματης επικοινωνίας. Η κατάσταση σε πραγματικό χρόνο του σκάφους και του εξοπλισμού που διαθέτει το ρομποτικό όχημα, παρακολουθούνται όλα από τον επίγειο σταθμό [15].



Εικόνα 1-5. Βασική αρχιτεκτονική ρομποτικού οχήματος επιφανείας [15].

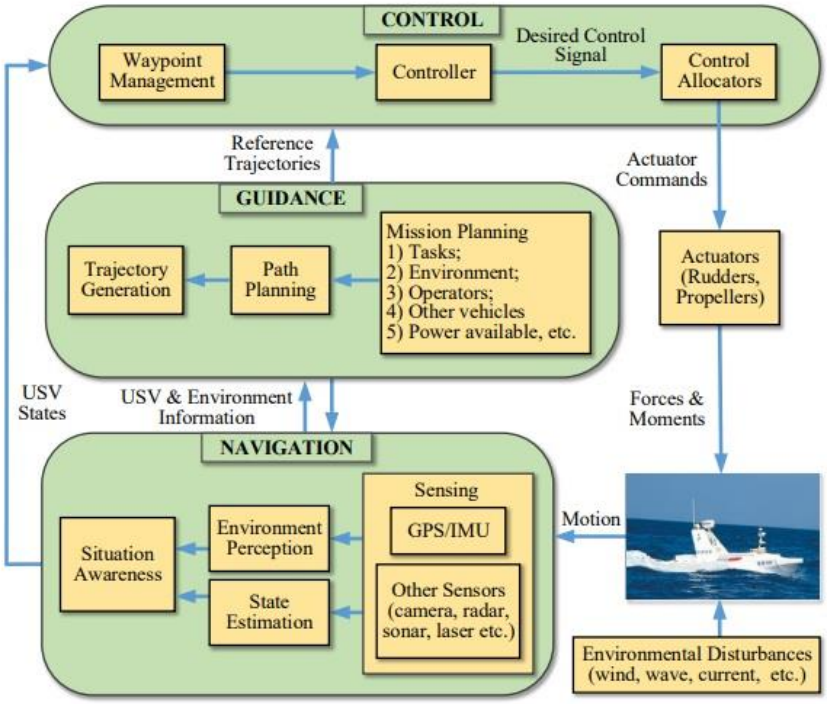
1.5 Γενική δομή των συστημάτων καθοδήγησης, πλοήγησης και ελέγχου ενός ρομποτικού οχήματος επιφανείας

Τα θεμελιώδη στοιχεία ενός ρομποτικού σκάφους είναι η καθοδήγηση, η πλοήγηση και ο έλεγχος. Τα υποσυστήματα αυτά λειτουργούν σε αλληλεπίδραση μεταξύ τους, σε σημείο που οι ατέλειες του ενός μπορούν να υποβαθμίσουν την απόδοση ενός άλλου. Ο ρόλος που επιτελεί το καθένα αναλύεται παρακάτω:

1. **Το υποσύστημα καθοδήγησης (GUIDANCE)**: είναι υπεύθυνο για τη συνεχή δημιουργία και ενημέρωση εντολών ομαλής, εφικτής και βέλτιστης τροχιάς στο σύστημα ελέγχου σύμφωνα με τις πληροφορίες που παρέχονται από το σύστημα πλοήγησης για τις ανατεθειμένες αποστολές, την ικανότητα του οχήματος και τις περιβαλλοντικές συνθήκες [15].
2. **Το υποσύστημα πλοήγησης (NAVIGATION)**: επικεντρώνεται στον εντοπισμό της τρέχουσας και μελλοντικής κατάστασης του οχήματος, όπως είναι η θέση, ο προσανατολισμός, η ταχύτητα, η επιτάχυνση αλλά και τη παρούσα και μελλοντική κατάσταση του περιβάλλοντός του με βάση τις προηγούμενες και τρέχουσες καταστάσεις του σκάφους, καθώς και περιβαλλοντικές πληροφορίες,

συμπεριλαμβανομένων των ωκεάνιων ρευμάτων και τη ταχύτητα ανέμου που λαμβάνονται από τους αισθητήρες του οχήματος [15].

3. **Το υποσύστημα ελέγχου (CONTROL)** : εστιάζει στον προσδιορισμό των κατάλληλων δυνάμεων ελέγχου και ροπών που θα δημιουργηθούν σε συνδυασμό με τις οδηγίες που παρέχονται από τα συστήματα καθοδήγησης και πλοήγησης, ενώ ταυτόχρονα ικανοποιεί τους επιθυμητούς στόχους ελέγχου [15].



Εικόνα 1-6. Γενική δομή των συστημάτων καθοδήγησης, πλοήγησης και ελέγχου ενός ρομποτικού οχήματος επιφανείας [15].

Τα υποσυστήματα που αναφέρθηκαν, συνθέτουν ουσιαστικά το πιο βασικό κομμάτι της αρχιτεκτονικής ενός αυτόνομου μη επανδρωμένου σκάφους, που είναι το **Guidance Navigation Control (GNC).** Με τη πρόοδο των ηλεκτρονικών το GNC αποτελείται από ένα ανεπτυγμένο ενσωματωμένο σύστημα που περιέχει μία ή περισσότερες μονάδες επεξεργασίας, αισθητήρες και διατάξεις εισόδων/εξόδων για την αλληλεπίδραση του με το περιβάλλον. Κύριος σκοπός της ενσωμάτωσης είναι η εξοικονόμηση χώρου και όγκου για την δημιουργία πιο αποδοτικών οχημάτων. Η μονάδα καθοδήγησης πλοήγησης και ελέγχου και το σύστημα τηλεπικοινωνιών ενός μη επανδρωμένου οχήματος παίζουν καθοριστικό ρόλο στο εύρος εφαρμογών και την προσαρμοστικότητα του.

ΚΕΦΑΛΑΙΟ 2

Θεωρητικό υπόβαθρο των τεχνολογιών που θα χρησιμοποιηθούν

2.1 Εισαγωγή

Στα πλαίσια της συνοχής και της πλήρης κατανόησης του περιεχομένου της παρούσας εργασίας, θα γίνει περιγραφή κάποιων βασικών εννοιών και τεχνολογιών που χρησιμοποιήθηκαν. Ο αναγνώστης θα έχει την ευκαιρία να επικαιροποιήσει τις γνώσεις του όσον αφορά τις τεχνολογίες IoT, τα πρωτόκολλα επικοινωνιών καθώς θα γίνει και παρουσίαση κάποιων υπηρεσιών που βοηθούν στην διασύνδεση τους.

2.2 Διαδίκτυο των Πραγμάτων

Συνεχώς, στα μέσα μαζικής ενημέρωσης και σε τεχνολογικά μέσα γίνεται αναφορά στο «Διαδίκτυο των πραγμάτων» ή αλλιώς «Internet of Things». Ο όρος συναντάται πολύ συχνά, σε εφαρμογές που αναφέρονται σε έξυπνα σπίτια, έξυπνη διαχείριση συσκευών και απομακρυσμένο έλεγχο. Αν επιμερίσουμε τον όρο, στην έννοια «πράγματα», μπορούν να ενταχθούν όλες οι συσκευές ή οντότητες που διαθέτουν δυνατότητες επικοινωνίας με το περιβάλλον. Η δυνατότητα διασύνδεση των συσκευών αυτών μεταξύ τους και η επικοινωνία τους με το διαδίκτυο ολοκληρώνει τον όρο «Διαδίκτυο των πραγμάτων» [16]. Ο όρος αναφέρθηκε πρώτη φορά από τον Kevin Ashton το 1999 σε ένα συνέδριο της Procter & Gamble για την ενσωμάτωση της τεχνολογίας RFID στην εφοδιαστική της αλυσίδα [17]. Από τα μέσα της δεκαετίας του 1980 έως την δεκαετία του 1990 είχε ήδη αρχίσει να αυξάνεται η χρήση πραγμάτων με δυνατότητες διασύνδεσης και απομακρυσμένης διαχείρισης. Η αύξηση αυτή οφειλόταν επίσης και στην μείωση του κόστους των υλικών δικτύωσης αλλά και στην σμίκρυνση αυτών των συσκευών και των αισθητήρων που χρησιμοποιώντουσαν σε αυτές τις εφαρμογές. Μία καλή ερμηνεία του διαδικτύου των πραγμάτων δίνεται από το [18] το οποίο το ορίζει ως «ένα πολύπλοκο συνδυασμό τεχνολογικών οικοσυστημάτων που διασυνδέουν τον φυσικό κόσμο με τον κόσμο του διαδικτύου, κάνοντας χρήση προηγμένων υπολογιστικών συσκευών και νεφοϋπολογιστικών υπηρεσιών» [18]. Βάση αυτού του ορισμού γίνεται αναφορά σε ένα δίκτυο το οποίο θα πρέπει να βασιστεί σε μία αρχιτεκτονική. Σύμφωνα με την «Διεθνή ένωση τηλεπικοινωνιών (ITU)» τα επίπεδα που στα οποία κατηγοριοποιείται η αρχιτεκτονική των IoT, θυμίζει τα επίπεδα δικτύωσης OSI και είναι:

- Το επίπεδο αισθητήρων
- Το επίπεδο Πρόσβασης
- Το επίπεδο δικτύου

- Το επίπεδο του Middleware[1]
- Τα επίπεδα εφαρμογών

Οι τεχνολογίες που απαρτίζουν το IoT είναι, τα πρωτόκολλα επικοινωνιών, τα μέσα ασύρματης ή ενσύρματης επικοινωνίας , οι ενεργοποιητές, οι αισθητήρες και περιβάλλοντα τεχνητής ή επαυξημένης νοημοσύνης. Καποιες από αυτές είναι το WiFi, το RFID, το NFC, το Bluetooth, το IP, Το Zigbee, το NBIoT και το MQTT. S [16].

2.3 Εικονικές Μηχανές Νέφους

Παλαιότερα, για να μπορέσει μία εταιρεία να συντηρήσει ένα κεντρικό server τοπικά, στον οποίο θα διατηρούσε και εκτελούσε κρίσιμες για αυτήν εφαρμογές, θα έπρεπε να επενδύσει ένα γενναίο χρηματικό ποσό σε αυτό. Ένα τέτοιο υπολογιστικό σύστημα, χρειάζεται:

- να έχει εξειδικευμένο προσωπικό για την συντήρηση του,
- να μπορεί να ενημερώνει συνεχώς το λογισμικό και τις εφαρμογές του,
- να το έχει συνεχώς σε λειτουργία που συνεπάγεται ενεργειακό κόστος,
- να επενδύει συνεχώς σε αναβάθμιση του υλικού μέρους του
- και να μπορεί είναι διαθέσιμο στο διαδίκτυο ανελλιπώς.

Μία «Εικονική Μηχανή Νέφους» ή αλλιώς «Cloud Virtual Machine», είναι ένα υπολογιστικό σύστημα οι πόροι του οποίου διατίθενται προς ενοικίαση από τον εκάστοτε πάροχο/εταιρεία. Η πρόσβαση, σε ένα τέτοιο υπολογιστικό σύστημα γίνεται απομακρυσμένα και είναι διαθέσιμο και λειτουργικό ανελλιπώς. Επίσης, η εταιρεία που προμηθεύει αυτά τις εικονικές μηχανές φροντίζουν για την συντήρηση, αναβάθμιση και την επίλυση τυχών προβλημάτων.

Στην παρούσα υλοποίηση έγινε χρήση μίας εικονικής μηχανής από τον Okeanos-knossos. Πρόκειται για μία υπηρεσία νέφους, η οποία υλοποιήθηκε από το GRNET, για να εξυπηρετεί ελληνικούς ακαδημαϊκούς φορείς όπως φοιτητές, καθηγητές και ερευνητές. Μέσω της δικτυακής πλατφόρμας του Cyclades, είναι δυνατή η δημιουργία, η διαχείριση των εικονικών μηχανών όπως επίσης και η σύνδεση σε αυτές μαζί με πολλές άλλες λειτουργίες. Όπως στις περισσότερες υπηρεσίες νεφοϋπολογιστικής έτσι και στο Cloud Virtual Machine του Okeanos-knossos διατίθενται τα παρακάτω χαρακτηριστικά:

- Επιλογή διαφορετικών λειτουργικών συστημάτων ανά εικονικό μηχάνημα
- Επιλογή των διαθέσιμων πόρων (επεξεργαστική ισχύς , RAM, χωρητικότητα κ.α.)
- Διασύνδεση με το διαδίκτυο
- Δυνατότητα δημιουργίας ιδιωτικών δικτύων
- Ενναλακτική πρόσβαση στα εικονικά μηχανήματα, πέραν του web browser
- Πολλαπλές επιλογές από Firewall αναλόγως των απαιτήσεων του χρήστη
- Δυνατότητα προσθήκη νέων αρχείων προς το εικονικό μηχάνημα
- Προσβασιμότητα μέσω RESful API [19].

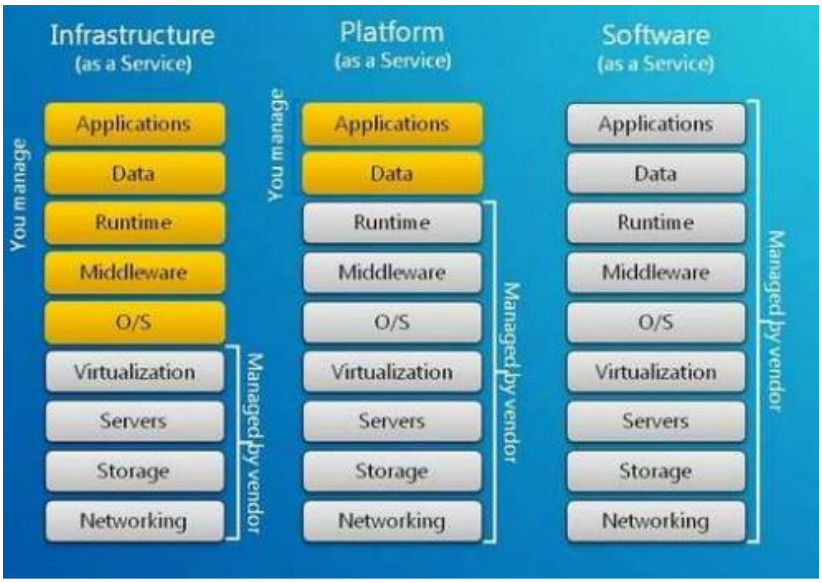
[1] Middleware: είναι αυτό το λογισμικό που αναλαμβάνει την διασύνδεση των επιπέδων δικτύου και εφαρμογών ή λειτουργικών συστημάτων.

2.4 Μοντέλα υπηρεσιών νέφους

Οι υπηρεσίες νέφους, δεν περιορίζονται μόνο στις εικονικές μηχανές, υπάρχουν υπηρεσίες που είτε στηρίζονται στην ύπαρξη μίας εικονικής μηχανής είτε υφίστανται ήδη και είναι διαθέσιμες για χρήση εντός μίας άλλης υπηρεσίας νέφους. Για τον διαχωρισμό αυτών των υπηρεσιών έχουν καθιερωθεί τρία μοντέλα:

- Η εφαρμογή ως υπηρεσία SaaS (Software as a Service)
- Η πλατφόρμα ως υπηρεσία PaaS (Platform as a Service)
- Και η υποδομή ως υπηρεσία IaaS (Infrastructure as a Service)

Ο τελικός χρήστης μπορεί να επέμβει σε συγκεκριμένα χαρακτηριστικά της υποδομής κάθε υπηρεσίας. Στην Εικόνα 2-1 φαίνονται τα επίπεδα υποδομής στα οποία μπορεί να επέμβει ο χρήστης ανάλογα με το μοντέλο της υπηρεσίας. ο κάθε ένα από αυτά τα μοντέλα έχει διαφορετικό



Εικόνα 2-1. Μοντέλα υπηρεσιών νέφους [20].

2.4.1 Software as a Service – SaaS

Το μοντέλο της εφαρμογής ως υπηρεσίας , αναφέρεται σε όλες αυτές τις εφαρμογές νέφους στις οποίες ο χρήστης έχει πρόσβαση μέσω διαδικτύου, μόνο στην κυρίως λειτουργία τους χωρίς να χρειάζεται να επέμβει σε κανένα επίπεδο της υποδομής τους. Οι SaaS εφαρμογές πολλές φορές έχουν την δυνατότητα αλληλεπίδρασης με άλλες εφαρμογές έστω και αν δεν το περιβάλλον τους ή πλατφόρμα τους είναι διαφορετική. Τέτοιες υπηρεσίες είναι: το Gmail, το Onedrive και το Google Docs [20].

2.4.2 Platform as a Service – PaaS

Το μοντέλο της πλατφόρμας ως Υπηρεσίας, είναι μία παραλλαγή του SaaS, μόνο που προσφέρει στον χρήστη το περιβάλλον ανάπτυξης μίας υπηρεσίας ή εφαρμογής. Συνήθως, μαζί με την πλατφόρμα προσφέρονται και κάποιοι υπολογιστικοί πόροι απαραίτητοι για την φιλοξενία της εφαρμογής που θα αναπτυχθεί. Οί πάροχοι αυτών των υπηρεσιών διαχειρίζονται την υποδομή της πλατφόρμας και προσφέρουν μία σειρά από εργαλεία που βοηθώντας τους προγραμματιστές να γίνουν πιο αποδοτικοί. Με τις PaaS εξοικονομείται

χρόνος και εργασία από την ανάπτυξη επιπλέον υποστηρικτικού κώδικα. Μερικές γνωστές πλατφόρμες είναι το Node-RED, το Google Apps Engine και το Heroku [21] [22].

2.4.3 Infrastructure as a Service – IaaS

Οι IaaS η αλλιώς η Υποδομή ως υπηρεσία αναφέρεται στους υπολογιστικούς πόρους που παρέχονται για την υποστήριξη είτε των PaaS είτε των SaaS. Κάθε υπηρεσία νέφους στηρίζεται σε μία υλικολογισμική υποδομή. Η υποδομή αυτή, προσφέρεται στον τελικό χρήστη και εκείνος με την σειρά του έχει την δυνατότητα να την παραμετροποιήσει ως προς τον διαμερισμό και τον όγκο των πόρων ενώ παράλληλα μπορεί να αποκτά πρόσβαση σε αυτή απομακρυσμένα βάση των εκάστοτε αναγκών του. Ο χρήστης έχει πλήρη πρόσβαση στην διαχείριση των πόρων που του προσφέρονται, στο λειτουργικό σύστημα, στην χωρητικότητα, στις εφαρμογές που θα εγκατασταθούν και πιθανότατα και σε κάποιες δικτυακές επιλογές. Τέτοια υπηρεσία προσφέρεται από το Okeanos-knossos, το Amazon Web Services (AWS) και το OpenStack [22].

2.5 Τεχνολογίες Επικοινωνιών

Στην παρούσα υλοποίηση ήταν απαραίτητη η ασύρματη διασύνδεση του σκάφους με τον χειριστή αλλά και το διαδίκτυο. Για να επιτευχθεί η επικοινωνία μεταξύ τους χρησιμοποιήθηκαν πρωτόκολλα επικοινωνίας IoT καθώς και εξειδικευμένα πρωτόκολλα που αφορούν την απομακρυσμένη παρακολούθηση και έλεγχο ενός σκάφους με σύστημα «Αυτόματου πιλότου».

2.5.1 WiFi

Εικόνα 2-2. Λογότυπο WiFi.

Το γνωστό σε όλους WiFi είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται για την ασύρματη διασύνδεση συσκευών και την ενοποίηση τους σε ένα τοπικό ασύρματο δίκτυο. Πιο συγκεκριμένα το WiFi υλοποιήθηκε αρχικά για να επιλύσει το πρόβλημα δικτύωσης συσκευών που βρίσκονταν σε χώρους ή περιοχές που ήταν δύσκολη έως αδύνατη η φυσική διασύνδεση τους ενσύρματα. Η WiFi Alliance είναι η ένωση, που ονοματοδότησε το WiFi και διατηρεί τα πνευματικά δικαιώματα του, το οποίο το συναντάμε και ως Wireless LAN ή αλλιώς WLAN. Είναι δηλαδή μία ασύρματη εκδοχή ενός ενσύρματου τοπικού δικτύου LAN και για τον λόγο αυτό βασίζεται και στο πρότυπο της IEEE 802.3. Το WiFi χρησιμοποιεί ραδιοκύματα (RF) για την μετάδοση των δεδομένων σε υψηλές ταχύτητες και αρχικά χρησιμοποιούσε μόνο την μπάντα των 2,4GHz (πρότυπο 802.11b). Οι δικτυακές ανάγκες κάθε σύνδεσης είναι διαφορετικές και οι ταχύτητες μεταφοράς δεδομένων ποικίλουν ανά περίπτωση. Για τον λόγο αυτό, στα πλαίσια της βελτίωσης τόσο της ταχύτητας όσο και της αδιάλειπτης σύνδεσης χωρίς παρεμβολές, αναπτύχθηκαν επιπλέον πρότυπα WiFi τα οποία πέραν της μπάντας 2,4GHz, χρησιμοποιούν και άλλες όπως αυτή των 5GHz. Τα πρότυπα

αυτά, βασίζονται στην μορφή 802.11X[2] ενώ η WiFi Alliance πλέον έχει αρχίσει να χρησιμοποιεί μία νέα προτυποποίηση με τη μορφή WiFi X[3]. Στον Πίνακα 2-1 φαίνονται ενδεικτικά κάποια από τα πρότυπα σε αντιστοίχιση με τις συχνότητες εκπομπής, τον ρυθμό μετάδοσης δεδομένων και την εμβέλεια εκπομπής τους [23].

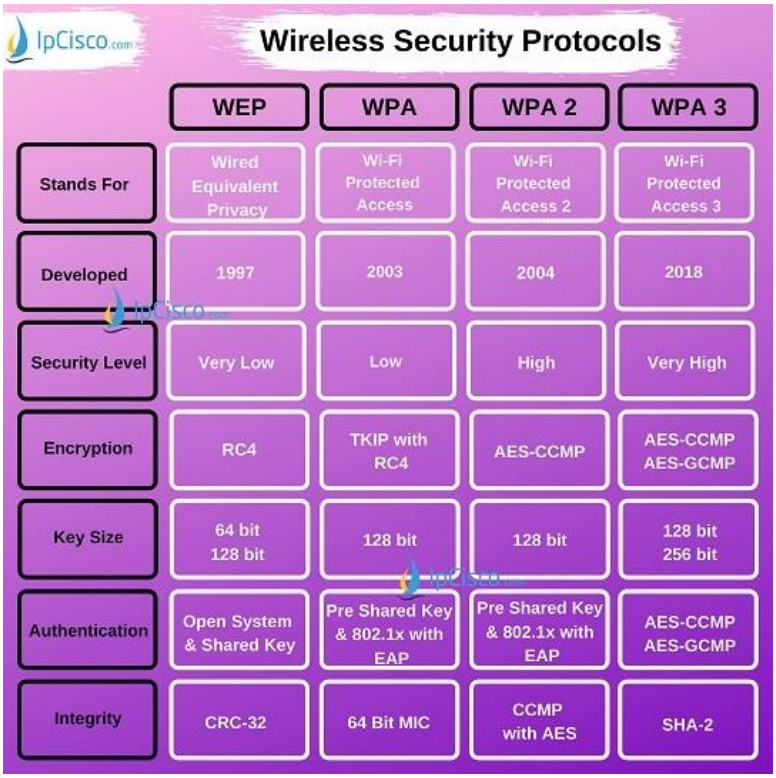
Πίνακας 2-1. Πρότυπα WiFi [24].

Πρότυπο	**Band**	**Ρυθμός Μετάδοσης**	**Εμβέλεια**
802.11b	2.4 GHz	Έως 11 Mbps	137m
802.11a	5 GHz	Έως 54 Mbps	121m
802.11g	2.4 GHz	Έως 54 Mbps	137m
802.11n ή WiFi 4	2.4/5 GHz	Έως 450 Mbps	251m
802.11ac ή WiFi 5	5 GHz	Έως 1300Mbps	304m
802.11ax ή WiFi 6	2.4/5 GHz	Έως 10Gbps	304m

Ο τρόπος λειτουργίας ενός δικτύου WiFi, είναι αρκετά απλός συνήθως εμπλέκεται ένα κεντρικό «Σημείο Πρόσβασης» ή «Access Point» (AP mode) και οι συσκευές που συνδέονται σε αυτό ως «Stations» (STA mode). Το Access Point είναι υπεύθυνο για την διευθυνσιοδότηση των συσκευών καθώς και την διασύνδεση τους στο διαδίκτυο αν αυτό είναι διαθέσιμο. Οι συσκευές που κάνουν χρήση του WiFi ολοένα και αυξάνονται καθώς σε αυτές προστέθηκαν, κινητά τηλέφωνα, φορητοί υπολογιστές και συσκευές IoT. Η διασύνδεση τόσο πολλών συσκευών θα πρέπει να γίνεται με ασφάλεια, έτσι έχουν αναπτυχθεί τρείς μέθοδοι κωδικοποίησης του τρόπου πρόσβασης και ταυτοποίησης των συσκευών σε κάθε δίκτυο WiFi. Αυτές οι μέθοδοι από την πιο ανασφαλή στην πιο ασφαλή είναι: η WEP, WPA-PSK (Pre Shared Key), WPA2 και WPA3 (χρήση κυρίως στο WiFi 6) [25].

[2] 802.11X: όπου X αντιστοιχεί το λατινικό αλβαβητικό σύμβολο κάθε προτύπου, όπως b,c, ac

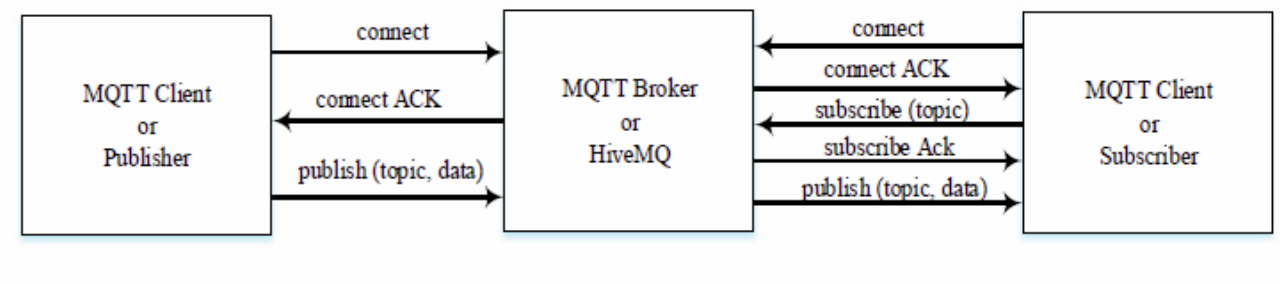
[3] WiFi X: όπου X αντιστοιχεί ένας αριθμός που υποδικνύει κάθε πρότυπο.



Εικόνα 2-3 Χαρακτηριστικά πρωτόκολλων ασφαλείας WiFi [26]

2.5.2 Πρωτόκολλο MQTT

Οι IoT συσκευές στην πλειονότητα τους είναι δικτυακές συσκευές με περιορισμένες δυνατότητες σε επεξεργαστική ισχύ. Η ανάγκη για επικοινωνία και διασύνδεση τέτοιων συσκευών οδήγησε στην υλοποίηση και χρήση εξειδικευμένων πρωτοκόλλων. Το MQTT (Message Queuing Telemetry Transport) είναι ένα ελαφρύ πρωτόκολλο επικοινωνίας M2M (Machine to machine) για IoT συσκευές. Το MQTT λειτουργεί έχοντας ως βάση το πρωτόκολλο TCP-IP αποστέλλοντας μηνύματα με την μέθοδο Subscribe/Publish σε πολλαπλές συσκευές. Η επικοινωνία μέσω MQTT αποτελείται από ένα MQTT broker, τους MQTT clients και τα topics των μηνυμάτων. Οι MQTT clients είναι οι συσκευές οι οποίες συμμετέχουν στο IoT δίκτυο και μπορούν να λαμβάνουν/αποστέλλουν μηνύματα στα topics τα οποία έχουν κάνει Subscribe (εγγραφή). Ο MQTT broker, που λειτουργεί σε μία διεύθυνση IP και συνήθως την πόρτα 1883, διαχειρίζεται τα μηνύματα που αποστέλλονται από τους clients στα διάφορα topics. Στην Εικόνα 2-4 φαίνεται πως διασυνδέονται οι Clients με τον broker. Συγκεκριμένα, ο client (publisher) δημιουργεί αίτημα σύνδεσης με τον broker και αφού γίνει αποδοχή της σύνδεσης μπορεί να αποστείλει κάποιο μήνυμα στο επιθυμητό topic. Ο client που επιθυμεί να λάβει τα μηνύματα ενός topic, πρέπει και αυτός με την σειρά του να κάνει αίτημα σύνδεσης στον broker, στη συνέχεια subscribe στο αντίστοιχο topic. Αφού λάβει τα μηνύματα επιβεβαίωσης μπορεί πλέον να λαμβάνει τα μηνύματα που αποστέλλονται σε αυτό. Ένας client μπορεί να είναι και Publisher και Subscriber δηλαδή και να λαμβάνει και να αποστέλλει μηνύματα. Επίσης, στον client δίνεται η δυνατότητα να επιλέξει αν θέλει ο broker να διατηρήσει το τελευταίο μήνυμα που απέστειλε μέσω του flag «Retain». Το μήνυμα παραμένει διαθέσιμο προς ανάγνωση, ακόμα και αν οι εγγεγραμμένοι client στο topic δεν είναι συνδεδεμένοι εκείνη την χρονική στιγμή. Ένα μήνυμα που έχει γίνει «Retain» μπορεί να σβηστεί αν αποσταλεί ένα νέο μήνυμα σε αυτό το topic ή αν αποσταλεί ένα κενό μήνυμα [25] [27].



Εικόνα 2-4 Διασύνδεση MQTT client -MQTT broker [27].

Η δομή ενός μηνύματος MQTT φαίνεται στην Εικόνα 2-5. Το QoS level αφορά την ποιότητα μετάδοσης του κάθε μηνύματος και έχει τρία επίπεδα το QoS 0, QoS 1 και QoS 2.

- Το QoS 0 αναφέρεται και ως «το πολύ μία», η αποστολή του μηνύματος θα γίνει μία φορά αλλά λόγω του ότι δεν απαιτείται επιβεβαίωση από τον παραλήπτη τότε το μήνυμα μπορεί και να χαθεί.
- Το QoS 1 αναφέρεται και ως «τουλάχιστον μία», η αποστολή του μηνύματος θα γίνει και θα απαιτηθεί επιβεβαίωση από τον παραλήπτη σε συγκεκριμένο χρονικό διάστημα. Αν δεν ληφθεί επιβεβαίωση, το μήνυμα θα σταλεί εκ νέου, οπότε μπορεί να αποσταλεί παραπάνω από μία φορές.
- Το QoS 2 αναφέρεται και ως «μόνο μία», η αποστολή του μηνύματος θα γίνει με τέτοιο τρόπο ώστε το μήνυμα να αποσταλεί μόνο μία φορά. Αυτό θα γίνει με την ανταλλαγή τεσσάρων πακέτων επιβεβαίωσης μεταξύ των εμπλεκομένων (τετραπλή χειραψία).

Format of fixed header.

Bit	7	6	5	4	3	2	1	0
Byte1	Message type				DUP flag	QoS level		RETAIN
Byte2	Remaining length							

Εικόνα 2-5 Δομή μηνύματος MQTT [25].

Από πλευράς ασφαλείας το πρωτόκολλο MQTT διαθέτει τρία επίπεδα. Το πρώτο βασίζεται στο ίδιο το δίκτυο μεταξύ broker και clients το οποίο θεωρείται ιδιωτικό. Το δεύτερο αφορά το επίπεδο της μετάδοσης των μηνυμάτων το οποίο χρησιμοποιεί τις μεθόδους κρυπτογράφησης TLS/SSL (Transport Layer Security/Secure Socket Layer). Το τρίτο αφορά το επίπεδο της εφαρμογής στην οποία για την ταυτοποίηση της κάθε συσκευής κάνει χρήση «ονόματος χρήστη» και «κωδικού χρήστη» [28].

2.5.3 Πρωτόκολλο MAVLink (Micro Air Vehicle Link)

Το MAVLink είναι ένα ελαφρύ πρωτόκολλο επικοινωνίας που αναπτύχθηκε κυρίως για την επικοινωνία μεταξύ ιπτάμενων αυτόνομων οχημάτων (drones) και του επίγειου σταθμού ελέγχου (GCS) καθώς και ως πρόσθετο σε εφαρμογές και άλλα λογισμικά. Πρόκειται για ένα πρωτόκολλο τηλεμετρίας ειδικά σχεδιασμένο για εφαρμογές που περιλαμβάνουν συστήματα με περιορισμένους πόρους ή εύρος μετάδοσης συχνοτήτων. Βασίζεται σε μία υβριδική αρχιτεκτονική “Publish/Subscribe” και “Point-to-Point” που παρέχει δύο μορφές μετάδοσης της πληροφορίας. Επομένως, οι ροές δεδομένων αποστέλλονται και αναρτιούνται (“published”) σε μορφή topics ενώ τα υπό-πρωτόκολλα που αφορούν την αποστολή και τις παραμέτρους του μη επανδρωμένου οχήματος, αποστέλλονται με αναμετάδοση “Point-to-Point”.

Η μετάδοση σε μορφή topics (“Publish/Subscribe”), εξοικονομεί εύρος ζώνης (bandwidth) αποστέλλοντας τα μηνύματα προς όλους τους subscribers, χωρίς να περιλαμβάνει εντός του μηνύματος πληροφορίες που αφορούν την ταυτότητα του σκάφους. Κάποια από αυτά τα μηνύματα που αποστέλλονται με αυτή τη μορφή είναι η θέση του οχήματος και το υψόμετρο.

Η μετάδοση σε μορφή point-to-point χρησιμοποιεί στοιχεία ταυτοποίησης του οχήματος στόχου όπως «ταυτότητα στόχου» (target ID) και «υποσύστημα στόχου» (target component). Κάνοντας χρήση των παραπάνω πεδίων, η μέθοδος αυτή μπορεί να εγγυηθεί την παράδοση του μηνύματος. Τα μηνύματα αυτά αφορούν αποκλειστικά το μη επανδρωμένο όχημα όπως εντολές, μηνύματα που αφορούν την αποστολή του, καθώς και διάφορες παραμετροποιήσεις που αφορούν την λειτουργία του.

Το πρωτόκολλο χάρη στην μικρή και απλή δομή του, υποστηρίζει πολλά επίπεδα μεταφοράς και μέσων μετάδοσης όπως τα πρωτόκολλα του διαδικτύου WiFi/Ethernet και τις ISM μπάντες μετάδοσης τηλεμετρίας στα 2.4 GHz (Bluetooth), 915 MHz, 868 MHz και 433 MHz. Επίσης για την ανάπτυξη του υποστηρίζονται πολλές γλώσσες προγραμματισμού, καθιστώντας το συμβατό με πολλούς γνωστούς μικροεπεξεργαστές και λειτουργικά συστήματα όπως ARM7, ATMega, STM32 και Windows, Linux, MacOS, Android και iOS αντίστοιχα.

Το MAVLink βασίζεται σε δυαδικά σειριακά μηνύματα, όπου το περιεχόμενο τους μετατρέπεται σε μια ακολουθία bytes που θα μεταδοθεί μέσα στο δίκτυο και ο παραλήπτης για να τα διαβάσει σωστά, θα πρέπει να αντιστρέψει την ακολουθία ανάγνωσης. Κάθε μήνυμα έχει προσαρτημένη μια επικεφαλίδα από bytes που περιέχουν πληροφορίες σχετικά με τον προορισμό, το παραλήπτη και άλλα δεδομένα σχετικά με τη πληροφορία που μεταφέρεται. Πιο συγκεκριμένα στην έκδοση MAVLink 1, ένα πλαίσιο μηνύματος είναι δομημένο σε ακολουθίες byte, ξεκινώντας σε μέγεθος από 8 bytes μόνο με την επικεφαλίδα χωρίς περιεχόμενο και φτάνοντας τα 263 bytes μαζί με το μέγιστο μέγεθος φέρουσας πληροφορίας, όπως φαίνεται στην Εικόνα 2-6.

Εικόνα 2-6. Δομή ενός πλαισίου μηνύματος MAVLink Version 1 [29].

Αντίστοιχα ο παρακάτω Πίνακας 2-2 επεξηγεί τμήματα ενός πλαισίου της έκδοσης MAVLink 1.

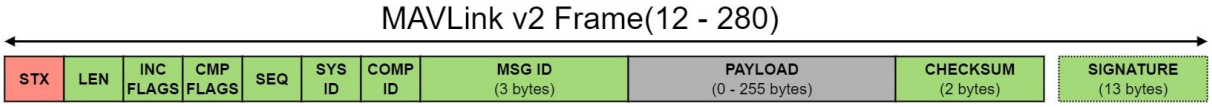
Πίνακας 2-2. Περιγραφή των πεδίων που συνθέτουν το πλαίσιο μηνύματος [29] [30].

Ακρωνύμιο	Μεγεθος	Εύρος τιμών	Περιγραφή
STX	1 Byte	0XFE	Ένα byte πληροφορίας που σηματοδοτεί την έναρξη ενός νέου πακέτου. 0XFE είναι η τιμή με την οποία ξεκινάνε όλα τα πακέτα MAVLink V1.
LEN	1 Byte	0 - 255	Δείχνει το μήκος σε bytes, της πληροφορίας (PAYLOAD) που εμπεριέχεται στο πακέτο.
SEQ	1 Byte	0 - 255	Δείκτης απώλειας πακέτων δεδομένων. Η τιμή του δείκτη SEQ αυξάνεται κατά 1 με κάθε αποστολή μηνύματος.

SYS ID	1 Byte	1 - 255	Η ταυτότητα του συστήματος (οχήματος) που στέλνει το μήνυμα. Χρησιμεύει για την διαφοροποίηση των συστημάτων στο δίκτυο. (Μέχρι 255 συστήματα ταυτόχρονα στο ίδιο δίκτυο)
COMP ID	1 Byte	1 - 255	Η ταυτότητα του εξαρτήματος που στέλνει το μήνυμα. Χρησιμεύει για την διαφοροποίηση των εξαρτημάτων σε ένα σύστημα, όπως για παράδειγμα ενός ελεγκτή πλοήγηση και της κάμερας ενός οχήματος.
MSG ID	1 Byte	0 - 255	Η ταυτότητα του τύπου μηνύματος που εμπεριέχεται στη πληροφορία (PAYLOAD). Χρησιμεύει για την αποκωδικοποίηση της πληροφορίας.
PAYLOAD	0 - 255 Bytes		Το περιεχόμενο του μηνύματος που εξαρτάται από τον τύπο του (MSG ID).
CHECKSUM	2 Bytes		Πληροφορία εντοπισμού λαθών, Checksum (low byte, high byte). CRC-16/MCRF4XX για το μήνυμα (εξαιρουμένου του STX byte). Περιέχει το CRC_EXTRA byte.

Σύμφωνα με τον παραπάνω Πίνακα 2-2, για την εξασφάλιση της ακεραιότητας ενός συστήματος πραγματοποιούνται δύο έλεγχοι. Ο πρώτος χρησιμοποιεί το CRC-16/MCRF4XX checksum για τον έλεγχο ακεραιότητας του προς μετάδοση πακέτου. Η δεύτερος επικυρώνει ότι τα στοιχεία ταυτοποίησης αντιστοιχούν με τα στοιχεία ταυτοποίησης από τα δεδομένα που ελήφθησαν κάνοντας χρήση του CRC-16-CCITT. Τα δεδομένα του CRC αποθηκεύονται σε μία μεταβλητή την CRC_EXTRA που χρησιμοποιείται βοηθητικά για να διασφαλιστεί ότι αποστολέας και παραλήπτης θα αντιληφθούν σωστά την διαμόρφωση του προς αποστολή μηνύματος. Παρόλο που δεν χρησιμοποιείται κάποιου είδους κρυπτογράφηση, η τελευταία μέθοδος πιστοποιεί με αυτό τον τρόπο ότι η πηγή μηνυμάτων είναι αξιόπιστη και πως η πληροφορίες δεν τροποποιήθηκαν κατά την μετάδοση .

Οι δυνατότητες του πρωτόκολλου επεκτάθηκαν με το MAVLink 2 καθώς προστέθηκαν επιπλέον πληροφορίες στην επικεφαλίδα του πλαισίου πληροφοριών, διατηρώντας παράλληλα τη συμβατότητα με τη προηγούμενη έκδοση (MAVLink Version 1). Η νέα δομή της επικεφαλίδας παρουσιάζεται στην Εικόνα 2-7.



Εικόνα 2-7. Το πλαίσιο ενός μηνύματος MAVLink Version 2 [29].

Πιο αναλυτικά, το μέγεθος της νέας επικεφαλίδας είναι 12 ή 27 bytes εάν προστεθούν τα 13 bytes ψηφιακής υπογραφής (“SIGNATURE”), που έχει σκοπό την ενίσχυση της ασφάλειας των δεδομένων του διαύλου επικοινωνίας, από κακόβουλες επιθέσεις. Επιπλέον στη νέα έκδοση προστέθηκαν δύο νέα bytes πληροφοριών “INC FLAGS” (incompatibility flags) και “CON FLAGS” (compatibility Flags).

Το byte “INC FLAGS” επηρεάζει και υποδεικνύει αλλαγές στη δομή του πλαισίου ενός μηνύματος, όπως για παράδειγμα η χρήση της προαιρετικής ψηφιακής υπογραφής. Εάν

εντοπιστεί διαφορά στη συμβατότητα από το παραλήπτη, κατά τον έλεγχο των δεικτών (flags) του πεδίου αυτού, τότε όλο το πακέτο του μηνύματος απορρίπτεται.

Το byte “CON FLAGS” δεν επηρεάζει την δομή του πλαισίου ενός μηνύματος και σκοπός του είναι να υποδείξει δείκτες στο περιεχόμενο του, που μπορούν να αγνοηθούν όταν δεν αναγνωρίζονται από το παραλήπτη. Τέτοιοί δείκτες για παράδειγμα, είναι ο βαθμός προτεραιότητας του πακέτου.

Τέλος το τμήμα της επικεφαλίδας “MSG ID” επεκτάθηκε από 1 σε 3 Bytes αυξάνοντας τους πιθανούς τύπους μηνυμάτων που μπορούν να αποκωδικοποιηθούν. Οι τύποι των μηνυμάτων χωρίζονται σε δύο κατηγορίες:

- Τα μηνύματα κατάστασης, που στέλνονται από το μη επανδρωμένο όχημα προς το σταθμό ελέγχου εδάφους ή άλλη συσκευή. Περιέχουν πληροφορίες σχετικά με τη κατάσταση του οχήματος όπως συντεταγμένες, προσανατολισμός, ταχύτητα και ενεργειακά αποθέματα.
- Τα μηνύματα εντολών, που στέλνονται από το σταθμό ελέγχου εδάφους ή άλλη εφαρμογή προς το μη επανδρωμένο όχημα, για την εκτέλεση ορισμένων ενεργειών ή τον έλεγχο της αποστολής του αυτόματου πιλότου.

Υπάρχει πληθώρα βιβλιοθηκών που υποστηρίζουν την ανάπτυξη εφαρμογών σχετικές με το πρωτόκολλο MAVLink. Για την βελτίωση των επιδόσεων σε χρόνο επεξεργασίας, RAM και άλλων χαρακτηριστικών έχει υλοποιηθεί και αναπτύσσεται συνεχώς μία ενναλακτική βιβλιοθήκη σε γλώσσα προγραμματισμού C, η fastMavlink[4]. Η βιβλιοθήκη αυτή είναι ιδανική για χρήση σε ενσωματωμένα συστήματα, καθώς εξοικονομεί προγραμματιστικό χρόνο και βελτιστοποιεί κάποιες εργασίες, παρέχοντας έτοιμες συναρτήσεις που αποφεύγουν τη χρήση προσωρινής μνήμης buffers ή επιπλέον καταχωρήσεις μεταβλητών [29] [30].

2.5.4 Συστήματα τηλεχειρισμού και το πρωτόκολλο ACCESS της Frsky

Τα συστήματα τηλεχειρισμού μοντέλων και μη επανδρωμένων οχημάτων με το πέρασμα του χρόνου έχουν υποστεί πολλές αλλαγές στο υλικό τους μέρος, αλλά και στο τρόπο λειτουργίας τους. Αρχικά, τα πρωτόκολλα ασύρματης μετάδοσης, ήταν αναλογικά φέροντα σήματα διαμορφωμένα χωρίς κωδικοποίηση [31]. Η εξέλιξη της τεχνολογίας και η ανάγκη για συστήματα τηλεχειρισμού που παρέχουν μεγαλύτερη αξιοπιστία, ασφάλεια και ανθεκτικότητα στις παρεμβολές, οδήγησαν στην υιοθέτηση των ISM μπαντών συχνοτήτων εκπομπής στα 868 MHz, 900 MHz και 2.4GHz [32]. Ειδικότερα, στις συχνότητες αυτές, αναπτύχθηκαν σύγχρονα πρωτόκολλα ασύρματης μετάδοσης αποτελούμενα από φέροντα σήματα με ψηφιακές διαμορφώσεις, που κάνουν χρήση τεχνικών εξάπλωσης φάσματος «Spread Spectrum» και κρυπτογράφησης της φέρουσας πληροφορίας [33]. Με βάση τα χαρακτηριστικά αυτά που αναφέρθηκαν προηγουμένως, κάθε εταιρία στο χώρο των τηλεκατευθύνσεων έχει αναπτύξει το δικό της κλειστό πρωτόκολλο τηλεπικοινωνιών, με διαφορετική Spread Spectrum τεχνική και κρυπτογράφησή, ώστε να μπορέσουν να εξασφαλίσουν το μερίδιο τους στην αγορά.

[4] olliw / fastmavlink Library: https://github.com/olliw42/fastmavlink

Εικόνα 2-8. Ένα τηλεχειριστήριο με τον δέκτη του από την εταιρεία FrSky.

Ένα ολοκληρωμένο σύστημα τηλεχειρισμού, αποτελείται από το τηλεχειριστήριο πομπό (TX) και το δέκτη (RX) όπως παρουσιάζεται στην Εικόνα 2-8. Ανάλογα την ανάγκη, το κάθε μοντέλο τηλεχειριστήριου έχει διαφορετικό αριθμό και τύπο εισόδων όπως διακόπτες, κουμπιά, ποτενσιόμετρα και joystick. Η μηχανική κατάσταση (θέση) αυτών των εξαρτημάτων εισόδου καταγράφεται και ομαδοποιείται σε κανάλια ελέγχου, η τιμή των οποίων, αποστέλλεται στον δέκτη (RX), μέσω των προαναφερθέντων πρωτόκολλων ασύρματης μετάδοσης. Ο δέκτης τοποθετείται στο μη επανδρωμένο όχημα και εξάγει τις τιμές των καναλιών σε ένα ή περισσότερα αναλογικά ή ψηφιακά RX πρωτόκολλα. Μπορεί για παράδειγμα, ένας δέκτης να χρησιμοποιεί την κλασσική αναλογική μέθοδο PWM, όπου κάθε κανάλι εξάγεται από ένα διαφορετικό pin του δέκτη σε μορφή ενός τετραγωνικού παλμού διαμορφωμένου κατά πλάτος. Σε άλλη περίπτωση, ένας δέκτης μπορεί να εξάγει τη θέση όλων των καναλιών χρησιμοποιώντας ένα ψηφιακό πρωτόκολλο μέσω μίας θύρας UART. Πολλά από τα συστήματα τηλεχειρισμού υποστηρίζουν αμφίδρομη επικοινωνία, επιτρέποντας τη μετάδοση δεδομένων τηλεμετρίας προς το πομπό (τηλεχειριστήριο), όπως μετρήσεις αισθητήρων ή άλλων ψηφιακών δεδομένων που καταγράφει ο δέκτης [34].

Η FrSky Electronic Co. δραστηριοποιείται στον χώρο της έρευνας και της ανάπτυξης ηλεκτρονικών και εφαρμογών που χρησιμοποιούνται σε συστήματα απομακρυσμένου και έξυπνου ελέγχου. Πιο συγκεκριμένα αναπτύσσει εξοπλισμό που χρησιμοποιείται ευρέως στην τηλεπικοινωνία, στα τηλεκατευθυνόμενα μοντέλα, την γεωργία και την βιομηχανία. Ο εξοπλισμός αυτός αφορά πομπούς, δέκτες, flight controllers, αισθητήρες καθώς και το ενσωματωμένο σε αυτά λογισμικό. Το πρωτόκολλο ACCESS (Advanced Communication Control Elevated Spread Spectrum) έχει αναπτυχθεί από την Frsky για αποκλειστική χρήση με τις τηλεκατευθύνσεις που εμπορεύεται. Το ACCESS βασίζεται σε μία παλαιότερη έκδοση το ACCST (Advanced Continuous Channel Shifting Technology) αλλά προσφέρει κάποια χαρακτηριστικά όπως το Smart Share, Smart Match, το Trio Control και βελτιωμένη απόδοση. Το Smart Share, δίνει την δυνατότητα αποθήκευσης των πιστοποιητικών διαφορετικών πομπών σε ένα δέκτη. Έτσι παρέχεται η δυνατότητα ελέγχου ενός μοντέλου από πολλά

διαφορετικά τηλεχειριστήρια, όπως για παράδειγμα από ένα εφεδρικό σε περίπτωση βλάβης. Το Smart Match προσφέρει διπλή ασφάλεια αφού χρησιμοποιεί διπλή πιστοποίηση μεταξύ δέκτη και πομπού. Με το Trio control το πρωτόκολλο ACCESS επιτρέπει σε ένα πομπό να συνδεθεί με έως και τρείς δέκτες ανά μοντέλο καθώς και τον έλεγχο ή την προβολή της τηλεμετρίας τους. Επομένως αυξάνεται η αξιοπιστία του συστήματος ελέγχου ενός μη επανδρωμένου οχήματος, καθώς εξασφαλίζονται εφεδρικοί δίαυλοι επικοινωνίας με το τηλεχειριζόμενο όχημα [35] [36]. Κάποια από τα χαρακτηριστικά του πρωτόκολλου ACCESS είναι ότι:

- Υποστηρίζει 24 κανάλια με αυξημένο ρυθμό μετάδοσης και μικρές καθυστερήσεις.
- Παρέχει τη δυνατότητα χρήσης του δέκτη του τηλεχειριστηρίου σαν αναλυτή φάσματος.
- Παρέχει λειτουργία μετρητή ισχύος του σήματος λήψης μέσω του τηλεχειριστήριου.
- Το πρωτόκολλο είναι συμβατό με τα αντίστοιχα συστήματα πομποδεκτών της FrSky στις ISM μπάντες εκπομπής των 868 MHz, 915MHz και 2.4GHz.
- Παρέχει τη δυνατότητα ενημέρωσης και παραμετροποίησης του υλικολογισμικού των δεκτών ασύρματα - OTA (over the air).
- Διαθέτει προηγμένο αλγόριθμο κρυπτογράφησης.

2.6 Το λογισμικό ελεγκτών πλοήγησης Ardupilot

Το Ardupilot είναι ένα ευέλικτο λογισμικό ανοιχτού κώδικα για αυτόματους πιλότους και αποτελεί τη καρδιά ενός μη επανδρωμένου οχήματος. Η ανάπτυξη του έχει ξεκίνησε το 2009 βασισμένο αρχικά στη πλατφόρμα μικροελεγκτή Arduino από όπου πήρε και το όνομα του. Από τότε συνεχώς ενημερώνεται και αναβαθμίζεται χάριν στην υποστήριξη του από μία μεγάλη κοινότητα προγραμματιστών και λάτρεις του αντικειμένου. Το υλικολογισμικό μπορεί να παραμετροποιηθεί για τον έλεγχο σχεδόν κάθε είδους οχήματος και ρομπότ όπως:

- Αεροσκάφη σταθερής πτέρυγας και VTOL (“Plane” firmware).
- Αερόστατα (οχήματα ελαφρύτερα του αέρα, “Blimp” firmware).
- Ελικόπτερα με ένα ή περισσοτέρους ρότορες (“Copter” firmware).
- Επίγεια rover (“Rover” firmware).
- Σκάφη επιφανείας (μηχανοκίνητα ή ιστιοφόρα) (“Rover” firmware).
- Ρομπότ ισορροπίας (“Rover” firmware).
- Υποβρύχια (“Sub” firmware).
- Ρομπότ προσανατολισμού κεραιών (“Antenna Tracker” firmware).

Το λογισμικό του Ardupilot έχει εξελιχθεί ώστε να είναι συμβατό με πάρα πολλές πλατφόρμες ενσωματωμένων συστημάτων, που απαρτίζονται από ένα ή περισσότερους μικροελεγκτές ή μικροεπεξεργαστές, βασισμένοι στην αρχιτεκτονική ARM. Τα συστήματα αυτά, είναι συνδεδεμένα με περιφερειακούς αισθητήρες που χρησιμοποιούνται για τη πλοήγηση του οχήματος. Μερικοί από τους αισθητήρες αυτούς μπορεί να είναι:

- Γυροσκόπια.
- Επιταχυνσιόμετρα.
- Μαγνητόμετρα (πυξίδες).
- Βαρόμετρα (υψομετρικά).
- Ταχύμετρα.
- Μετρητές απόστασης.
- Δέκτες γεωδαιτικών δεδομένων (GNSS)

- Κάμερες.

Το σύνολο ή μέρος των αισθητήρων αυτών μπορεί να ενσωματώνονται ή να συνδέονται εξωτερικά της ηλεκτρονικής διάταξης του ελεγκτή πλοήγησης που εμπεριέχει το υλικολογισμικό Ardupilot. Ένας συμβατός ελεγκτής πλοήγησης, περιέχει επίσης δίαυλους επικοινωνίας SPI, I2C, CAN BUS και UART, για την επικοινωνία του με περιφερειακές συσκευές όπως ασύρματους πομποδέκτες, αισθητήρες και ενεργοποιητές. Επίσης μπορεί να ενσωματώνει ψηφιακές εξόδους για τον άμεσο έλεγχο κινητήρων και ενεργοποιητών μέσω PWM ή άλλων ψηφιακών πρωτοκόλλων όπως το DShot [37]. Βασικό πλεονέκτημα είναι η υποστήριξη συνοδευτικών υπολογιστών συστημάτων γνωστών και ως "Companion Computers". Επομένως, χάρη στο πρωτόκολλο επικοινωνίας MAVLink ένας ελεγκτής πλοήγησης Ardupilot μπορεί να συνεργαστεί με συστήματα όπως τα Raspberry Pi, τα Nvidia Jetson και άλλες υπολογιστικές πλατφόρμες, ανοίγοντας τους ορίζοντες για νέες δυνατότητες και εφαρμογές στα μη επανδρωμένα οχήματα που το υποστηρίζουν [38].

Το MAVLink, επιτρέπει στη δημιουργία πληθώρας εφαρμογών (Software) για τη διαχείριση και την ανταλλαγή δεδομένων τηλεμετρίας με το μη επανδρωμένο όχημα. Το Mission Planner είναι ένα πρόγραμμα GCS που έχει δημιουργηθεί για αυτό το σκοπό, επιτρέποντας την παραμετροποίηση, έλεγχο και παρακολούθηση του οχήματος, καθώς και το σχεδιασμό της αποστολής που θα εκτελέσει το σύστημα πλοήγησης [39].

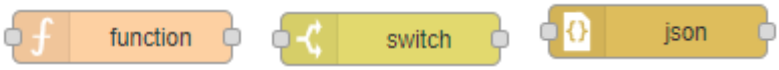
2.7 Η πλατφόρμα (PaaS) Node-RED

Εικόνα 2-9. Λογότυπο Node Red.

Το Node-RED είναι μία πλατφόρμα ως υπηρεσία ανοιχτού κώδικα που χρησιμοποιείται σε υλοποιήσεις που αφορούν το Διαδίκτυο των Πραγμάτων και επικεντρώνεται στην απλοποίηση της διασύνδεσης των διαφόρων συσκευών μεταξύ τους. Υλοποιήθηκε από την IBM με βάση την πλατφόρμα Node.js που κάνει χρήση της γλώσσας προγραμματισμού javascript. Το Node-RED δεν χρησιμοποιεί εντολές για την διασύνδεση των συσκευών καθώς βασίζεται σε ένα είδος οπτικοποιημένου προγραμματισμού. Οι διάφορες λειτουργίες ή παραμετροποιήσεις των δεδομένων γίνεται μέσω κάποιων «nodes» (κόμβων) και διασύνδεση μεταξύ τους γίνεται με εικονικά καλώδια «wires» τα οποία δημιουργούν «flows» (ροές δεδομένων), από την έξοδο του ενός κόμβου στην είσοδο ενός άλλου ή άλλων. Τα μηνύματα που μεταφέρονται μεταξύ των κόμβων είναι της μορφής msg.payload που οφείλεται στην javascript και αναλόγως της επεξεργασίας που θα υποστούν προβάλουν την αντίστοιχη έξοδο. Παρόλα αυτά στα μηνύματα μπορούν να προστεθούν και επιπλέον παράμετροι όπως msg.topic ή να αλλαχθεί η μορφή τους (π.χ. object, json, html). Επομένως, υπάρχουν:

- κόμβοι που λειτουργούν ως είσοδοι δεδομένων, (π.χ. inject, mqtt in). Κάποιοι από αυτούς διαθέτουν και κουμπί ενεργοποίησης για να αποστέλουν χειροκίνητα μία τιμή εισόδου.

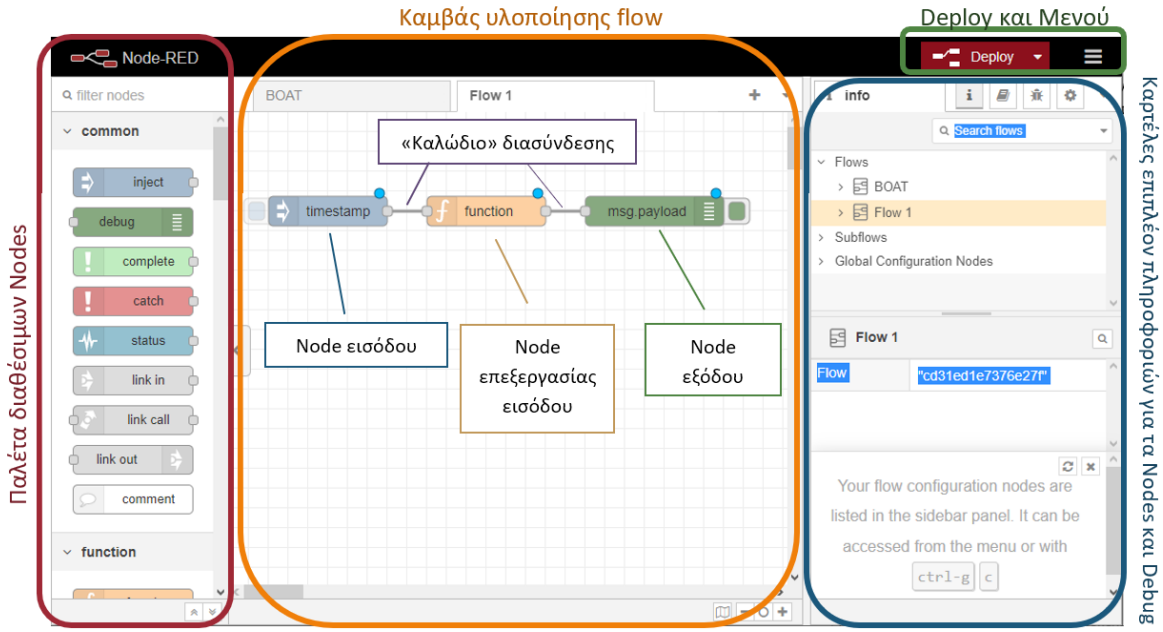
- κόμβοι που επεξεργάζονται τα δεδομένα και διαθέτουν και είσοδο και έξοδο (π.χ. function, switch, json)



- κόμβοι που προσφέρουν μόνο έξοδο (π.χ. debug, mqtt out)

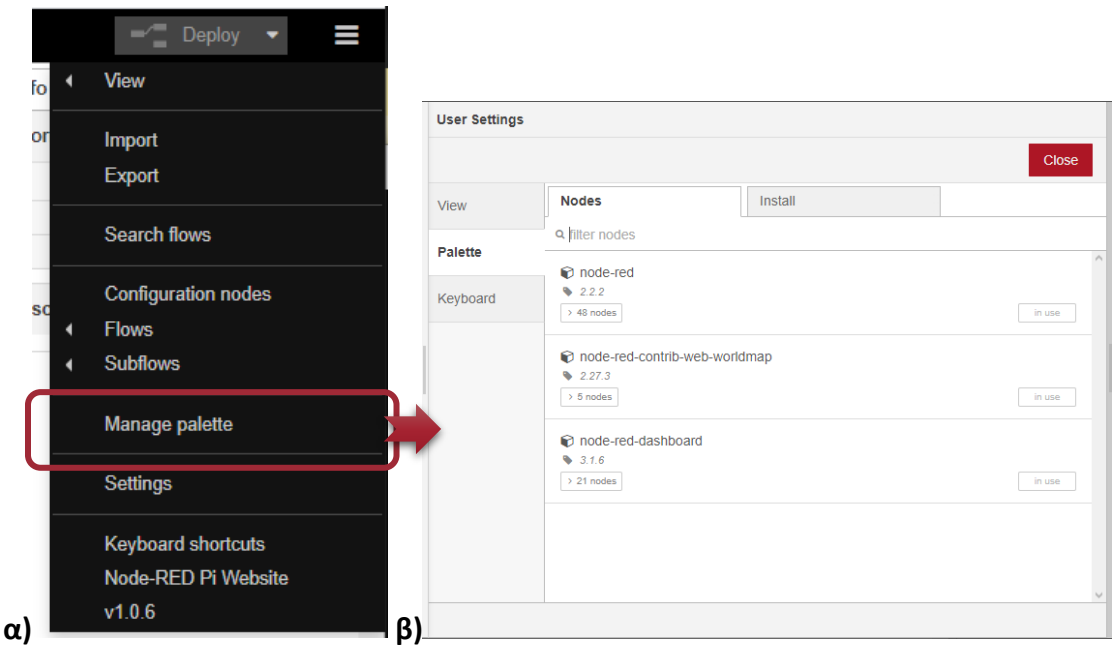
- κόμβοι για τοποθετηση σχολίων. (π.χ. comment)

Το Node-RED διαθέτει γραφικό περιβάλλον το οποίο είναι ευκόλως προσβάσιμο από ένα internet browser. Στο γραφικό περιβάλλον αυτό όπως φαίνεται και στην Εικόνα 2-10 από αριστερά προς τα δεξιά φαίνονται η πλαϊνή γραμμή εργαλείων με τα διαθέσιμα «Nodes» από τις εγκατεστημένες βιβλιοθήκες. Στη συνέχεια, εμφανίζεται ο καμβάς εργασίας στον οποίο δημιουργούνται τα «flows». Στα δεξιά υπάρχει η επιλογή «Deploy» με την οποία ενεργοποιούνται και εκτελούνται τα διαθέσιμα flows , το μενού επιπλέον επιλογών που αφορούν την παραμετροποίηση του Node-RED και οι καρτέλες με επιπλέον επιλογές που αφορούν τα διαθέσιμα «Nodes» και τα μηνύματα debugging.



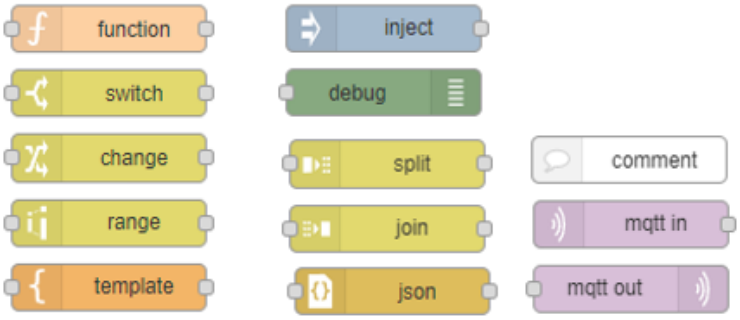
Εικόνα 2-10. Γραφικό Περιβάλλον Node-RED.

Η παλέτα στα αριστερά του API διαθέτει κάποια βασικά nodes, αλλά υπάρχει η δυνατότητα εγκατάστασης επιπλέον nodes που έχουν αναπτυχθεί από την κοινότητα ανάπτυξης του Node-RED. Από το μενού επιπλέων ρυθμίσεων του Node-RED (πάνω δεξιά) (Εικόνα 2-11 α) αν επιλεγεί το μενού «Manage Palette», ο χρήστης έχει την δυνατότητα να εμφανίσει και να εγκαταστήσει επιπλέον nodes ή να κάνει ενημέρωση (update) των ήδη εγκατεστημένων (Εικόνα 2-11 β).

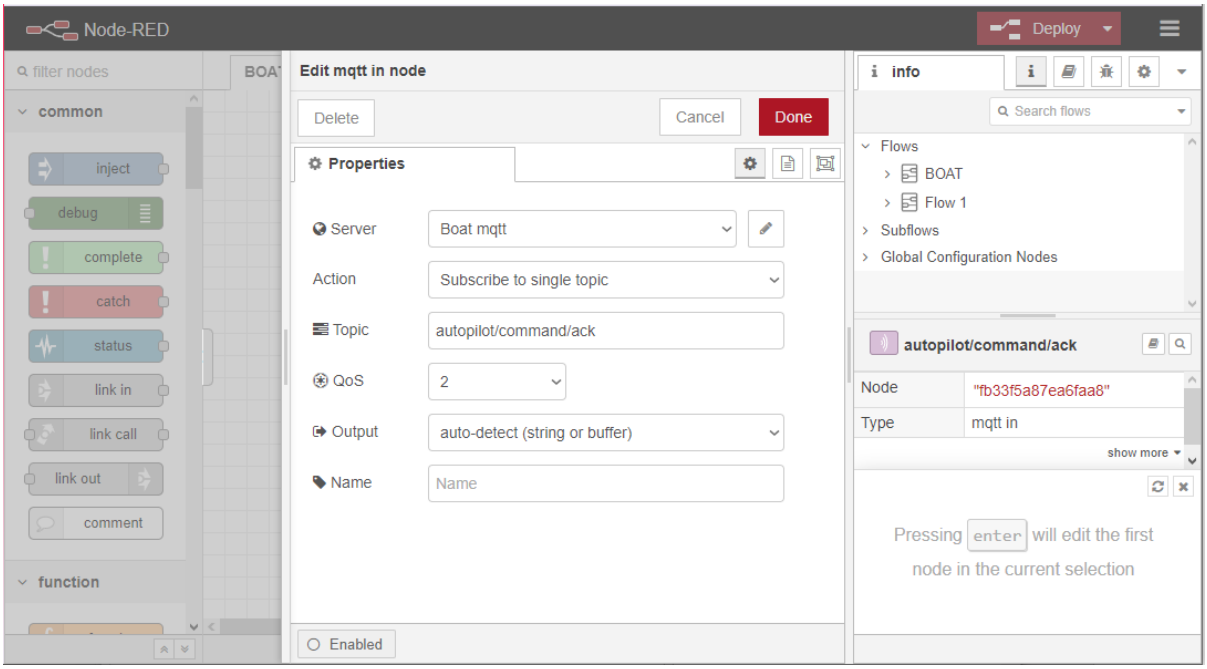


α) **β)**

Εικόνα 2-11. α) Μενού επιπλέων επιλογών και παραμετροποίησης Node-RED β) Manage Palette.



Εικόνα 2-12 Συχνά χρησιμοποιούμενοι κόμβοι.



Εικόνα 2-13. Δείγμα επιπλέον επιλογών για τον κόμβο του MQTT in.

Οι κόμβοι του Node-RED διαθέτουν και επιπλέον ρυθμίσεις στις οποίες ο χρήστης αποκτά πρόσβαση κάνοντας διπλό κλίκ πάνω τους (Εικόνα 2-13). Με αυτό τον τρόπο είναι δυνατή η περαιτέρω παραμετροποίηση και προγραμματισμός των λειτουργιών των κόμβων έτσι ώστε να εξυπηρετούν τις εκάστοτε ανάγκες της υλοποίησης. Πολλές φορές ο συνδυασμός πολλών κόμβων μπορεί να γίνει αρκετά πολύπλοκος και να απαιτεί πολύ χώρο στον καμβά. Για τον λόγο αυτό υπάρχει η δυνατότητα δημιουργίας Subflow τα οποία είναι κάποια βοηθητικά flow που δημιουργεί ο χρήστης και τρέχουν στο παρασκήνιο ενώ το αποτέλεσμα/έξοδος τους μπορεί να χρησιμοποιηθεί στο κυρίως flow. Το Node-RED μπορεί να δημιουργήθηκε αρχικά για υλοποιήσεις IoT, αλλά χάρη στην ευελιξία του και την επεκτασιμότητα του, το κάνει ιδανικό και για άλλες χρήσεις και εφαρμογές. Μπορεί επίσης να συνεργαστεί με άλλες πλατφόρμες, βάσεις δεδομένων, διαδικτυακές εφαρμογές καθώς και να αποστείλει ειδοποιήσεις στον χρήστη ή κάποια άλλη υπηρεσία [40] [41].

ΚΕΦΑΛΑΙΟ 3

Ανάπτυξη συστήματος

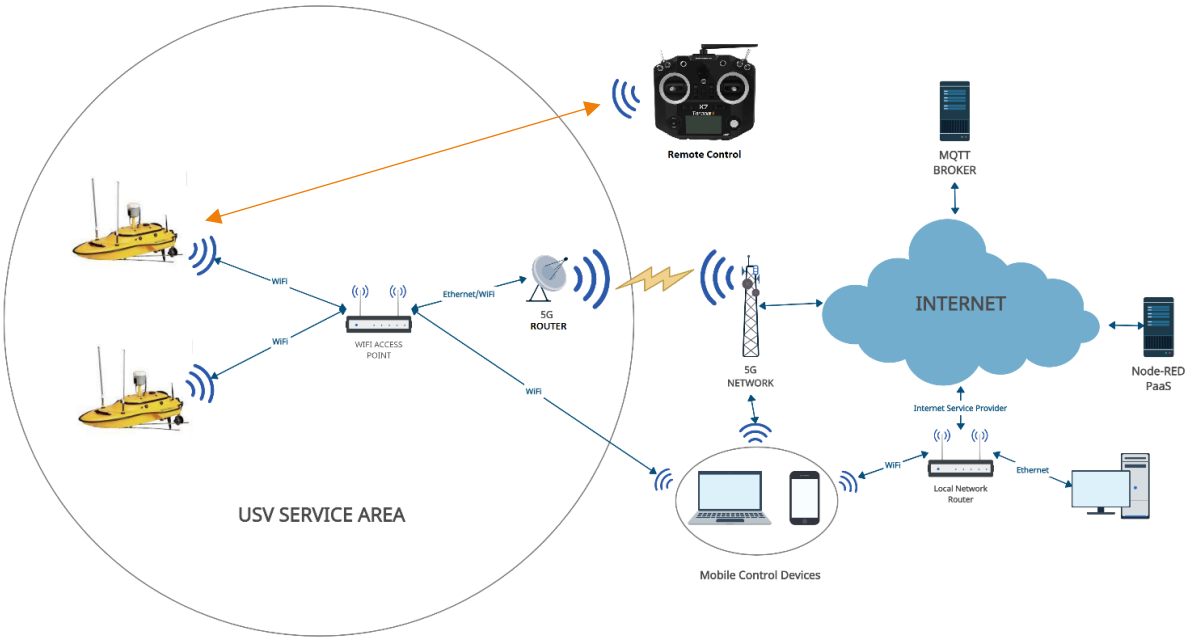
3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει η σχεδίαση και η ανάπτυξη του μη επανδρωμένου σκάφους επιφανείας. Θα παρουσιαστεί η αρχιτεκτονική στην οποία βασίζεται το ρομποτικό όχημα και θα αναλυθεί ο τρόπος που τα υποσυστήματα διασυνδέονται μεταξύ τους. Τέλος θα περιγραφεί η ανάπτυξη του μηχανολογικού, ηλεκτρονικού και προγραμματιστικού μέρους ξεχωριστά.

3.2 Η αρχιτεκτονική του συστήματος ελέγχου

Στην ενότητα αυτή περιγράφεται η αρχιτεκτονική στην οποία βασίζεται η λειτουργία του μη επανδρωμένου σκάφους επιφανείας. Η σχεδίαση του συστήματος χωρίστηκε σε δύο μέρη. Το πρώτο μέρος έχει να κάνει με την δομή του δικτύου μετάδοσης δεδομένων και ελέγχου του μη επανδρωμένου οχήματος. Στο δεύτερο μέρος αναλύεται η δομή των υποσυστημάτων του ρομποτικού οχήματος.

Όπως έχει αναφερθεί ο έλεγχος του συστήματος θα βασιστεί σε τεχνολογίες του διαδικτύου των πραγμάτων για αυτό το λόγο, η υλοποίηση κάνει χρήση ευρέως διαδεδομένων μέσων τηλεπικοινωνιών που στηρίζονται στο πρωτόκολλο διευθυνσιοδότησης IP. Στην Εικόνα 3-1 παρουσιάζεται η αρχιτεκτονική του δικτύου στο οποίο στηρίζεται η υλοποίηση.



Εικόνα 3-1. Αρχιτεκτονική δικτύου.

Αρχικά, σύμφωνα με το διάγραμμα της παραπάνω εικόνας φαίνεται ότι η επικοινωνία του μη επανδρωμένου οχήματος με το υπόλοιπο δίκτυο υποδομής βασίζεται στο ασύρματο πρωτόκολλο WiFi. Στα πλαίσια της εργασίας η χρήση του WiFi είναι η καλύτερη επιλογή καθώς, είναι ήδη ενσωματωμένο στις αναπτυξιακές πλατφόρμες μικροελεγκτών (ESP32 και ESP8266) που είναι εγκατεστημένες στο σκάφος της παρούσας υλοποίησης, βοηθώντας στη μείωση του κόστους υλοποίησης. Επιπλέον η χρήση ενός πρωτοκόλλου ασύρματης μετάδοσης δεδομένων, που βασίζεται στο πρωτόκολλο του διαδικτύου IP παρέχει την ευελιξία για εύκολη μετατροπή και επέκταση του συστήματος με άλλα συμβατά μέσα επικοινωνίας. Για λόγους ασφαλείας το μη επανδρωμένο σκάφος επιφανείας διαθέτει ένα δεύτερο μέσο ελέγχου, μέσω τηλεχειριστηρίου γενικής χρήσεως που έχει σχεδιαστεί για τον έλεγχο τηλεκατευθυνόμενων ιπτάμενων μοντέλων. Από το τηλεχειριστήριο μπορεί να παρακαμφθεί οποιαδήποτε αυτοματοποιημένη λειτουργία του οχήματος εξασφαλίζοντας την αποφυγή ατυχημάτων ανά πάσα στιγμή. Η μπάντα συχνοτήτων του συστήματος τηλεχειρισμού λειτουργεί στα 868 MHz και με αυτό τον τρόπο εξασφαλίζεται ότι δεν θα παρεμβάλλεται στη συχνότητα εκπομπής του WiFi. Στο σημείο πρόσβασης (AP: Access Point) που χρησιμοποιεί το μη επανδρωμένο όχημα, είναι συνδεδεμένος ένας δρομολογητής (Router) που δίνει στο τοπικό δίκτυο πρόσβαση στο διαδίκτυο (Internet), μέσω του δικτύου κινητής τηλεφωνίας (5G). Επιπλέον πρόσβαση στο διαδίκτυο έχει ο διακομιστής του πρωτόκολλου επικοινωνίας MQTT και η πλατφόρμα υπηρεσιών (IoT PaaS) Node-RED, που στήθηκαν στην ακαδημαϊκή υπηρεσία νέφους Okeanos-knossos. Ο διακομιστής MQTT (MQTT broker) είναι υπεύθυνος για τη διανομή δεδομένων μεταξύ του σκάφους και της πλατφόρμας υπηρεσιών Node-RED. Αντίστοιχα, η πλατφόρμα Node-RED συγκεντρώνει σε ένα γραφικό περιβάλλον όλες τις πληροφορίες που αφορούν το μη επανδρωμένο σκάφος επιφανείας. Αυτό επιτρέπει την διαχείριση του σκάφους, με τη χρήση ενός υπολογιστικού συστήματος, από οποιοδήποτε μέρος του κόσμου όπου υπάρχει πρόσβαση στο διαδίκτυο.

3.3 Μηχανολογικό Μέρος.

Στην ενότητα αυτή, περιγράφονται οι διεργασίες που πραγματοποιήθηκαν για να ετοιμαστούν τα μηχανολογικά μέρη του σκάφους επιφανείας που χρησιμοποιήθηκε στην εργασία αυτή. Για τη σωστή λειτουργία του, ένα τέτοιο όχημα θα πρέπει να έχει λειτουργικό πηδάλιο, σύστημα πρόωσης και στεγανό κύτος. Επιπλέον το σκάφος είναι απαραίτητο να διαθέτει τους κατάλληλους χώρους στο εσωτερικό του για την άνετη και ασφαλή τοποθέτηση των ηλεκτρονικών και ηλεκτρομηχανικών του εξαρτημάτων. Η πρόσβαση στο εσωτερικό του κύτους για οποιαδήποτε τροποποίηση ή συντήρηση των υποσυστημάτων του θα πρέπει να γίνεται με στεγανές θυρίδες. Για τις ανάγκες της παρούσας διπλωματικής επιλέχθηκε ένα ευρύχωρο πολυεστερικό μοντέλο ταχύπλοου σκάφους μήκους ενός (1) μέτρου όπως φαίνεται στην Εικόνα 3-2 .

Εικόνα 3-2. Το μοντέλο του σκάφους στη τελική του μορφή ύστερα από τις διεργασίες συντήρησης και τροποποιήσεων.

3.3.1 Αδιαβροχοποίηση

Η αδιαβροχοποίηση ήταν η πρώτη και σημαντικότερη διεργασία καθώς εξασφαλίζει την στεγανότητα του θαλάσσιου οχήματος και την αποφυγή ζημιών στον εξοπλισμό που θα εγκατασταθεί στο εσωτερικό του. Αξίζει να τονισθεί ότι το σκάφος που χρησιμοποιήθηκε ήταν μεταχειρισμένο και γι’ αυτό ο έλεγχος της στεγανοποίησης ήταν απαραίτητος. Αρχικά ελέγχθηκε το πολυεστερικό κύτος του σκάφους και διαπιστώθηκε ότι δεν υπήρχαν φθορές που θα μπορούσαν να προκαλέσουν εισροή υδάτων, όμως η θυρίδα πρόσβασης δεν ήταν υδατοστεγής. Οι λόγοι εισροής υδάτων οφείλονταν στον συνδυασμό των παρακάτω αίτιων:

- Έλλειψης μονωτικού υλικού γύρω από την περιμετρική επιφάνεια (κάσα) της θυρίδας, η οποία βιδώνεται στο κατάστρωμα του σκάφους,
- Φθορά στη λαστιχένιά φλάντζα στεγανοποίησης της θυρίδας,
- Η επιφάνεια του πολυεστερικού καταστρώματος στην οποία βιδώνεται η κάσα της θυρίδας, δεν ήταν επίπεδη, με αποτέλεσμα η γεωμετρία της πόρτας να αλλοιώνεται κατά την τοποθέτηση της.

Αν και τα παραπάνω αίτια που περιεγράφηκαν δεν είναι προφανή με μία πρώτη ματιά από κάποιον που δεν έχει την σχετική εμπειρία, παρόλα αυτά, σταδιακοί κύκλοι πειραματισμών μικρής βύθισης του σκάφους στο νερό και επιδιορθώσεων, έδειξαν αμέσως τα σημεία στα οποία υπήρχαν προβλήματα. Στην Εικόνα 3-3 φαίνονται τα σημεία που ευθύνονται για την έλλειψη στεγανότητας.

Για την επισκευή των προβλημάτων που σχετίζονται με τη θυρίδα ακολουθήθηκαν τα ακόλουθα βήματα με τη σειρά:

1. Ανακατασκευή της επιφάνειας του καταστρώματος πάνω στην οποία τοποθετείται η θυρίδα. Κύριος στόχος της μετατροπής ήταν να γίνει το κατάστρωμα στη περιοχή αυτή πιο άκαμπτο, στιβαρό και επίπεδο χωρίς εσοχές. Τα υλικά που χρησιμοποιήθηκαν ήταν ξύλο από κόντρα πλακέ θαλάσσης και πολυεστέρας ενισχυμένος με ίνες γυαλιού. Έτσι η επιφάνεια όπως παρουσιάζεται στην Εικόνα 3-3α τροποποιήθηκε στο τελικό αποτέλεσμα που φαίνεται στα αποσπάσματα της Εικόνα 3-4
2. Η τροποποιημένη περιοχή καθαρίστηκε από υπολείμματα πολυεστέρα και λειάνθηκε έως ότου γίνει επίπεδη.
3. Τοποθετήθηκε η νέα θυρίδα πρόσβασης με το απαραίτητο υλικό στεγανοποίησης και βάφτηκε περιμετρικά η περιοχή με πολυεστερικό χρώμα, για προστασία του πολυεστερικού καταστρώματος από την υπεριώδη ακτινοβολία.

Το τελικό αποτέλεσμα του καταστρώματος με τη πόρτα για πρόσβαση στο εσωτερικό του σκάφους απεικονίζεται στην Εικόνα 3-5. Με τις διεργασίες αυτές πλέον είχε εξασφαλιστεί η στεγανότητα του σκάφους, γεγονός που αποδείχθηκε από πειραματισμούς που πραγματοποιήθηκαν με μικρές βυθίσεις του οχήματος στο νερό.

Στο στάδιο αυτό οι διεργασίες αδιαβροχοποίησης των αξόνων του πηδαλίου και της προπέλας δεν πραγματοποιήθηκαν και επομένως δεν ελέγχθηκαν για τη στεγανότητα τους. Αυτό οφείλεται στο γεγονός ότι πρώτα έπρεπε να διερευνηθούν οι παράμετροι που σχετίζονται με τους μηχανισμούς του κάθε υποσυστήματος ξεχωριστά. Τέτοιοι παράμετροι μπορεί να είναι μηχανολογικά εξαρτήματα, όπως σύνδεσμοι μετάδοσης κίνησης και οι φθορές αλλά και οι μετατροπές που μπορεί να προκύψουν κατά τη χρήση τους είναι δύσκολο να προβλεφθούν εξ αρχής. Ως εκ τούτου, οι οπές στο κύτος του σκάφους που σχετίζονται με τα συστήματα αυτά, μονώθηκαν έτσι ώστε να μελετηθούν ξεχωριστά σε επόμενη φάση.

α)

Παραπάνω φαίνεται η επιφάνεια του καταστρώματος στην οποία βιδώνεται η επιφάνεια στήριξης. Ύστερα από έλεγχο, διαπιστώθηκε ότι η επιφάνεια δεν ήταν επίπεδη, με αποτέλεσμα η γεωμετρία της θυρίδας να αλλοιώνεται.

β)

Παραπάνω απεικονίζεται το αποσυναρμολογημένο πλαίσιο της θυρίδας, όπου φαίνεται η έλλειψη μονωτικού υλικού.

γ)

Παραπάνω φαίνεται στη περίμετρο της θυρίδας ως μαύρη λωρίδα, η λαστιχένια φλάντζα στεγανοποίησης. Η φλάντζα λόγω ηλικίας είχε αλλοιωθεί και η στεγανοποίηση δεν ήταν πλέον εφικτή.

Εικόνα 3-3. Τα προβλήματα της θυρίδας πρόσβασης στο εσωτερικό του σκάφους.

α)

Τοποθέτηση ξύλινου πλαισίου για το γέμισμα και την ενίσχυση της περιοχής της εσοχής, όπου στηριζόταν η θυρίδα προηγουμένως.

β)

Αφού τοποθετήθηκε το ξύλινο πλαίσιο, στη συνέχεια η περιοχή γεμίστηκε και επικαλύφθηκε με υγρό πολυεστέρα ενισχυμένο από ίνες γυαλιού.

γ)

Τελικό αποτέλεσμα μετατροπής της περιοχής του καταστρώματος, όπου στηρίζεται η θυρίδα πρόσβασης. (κοντινό πλάνο)

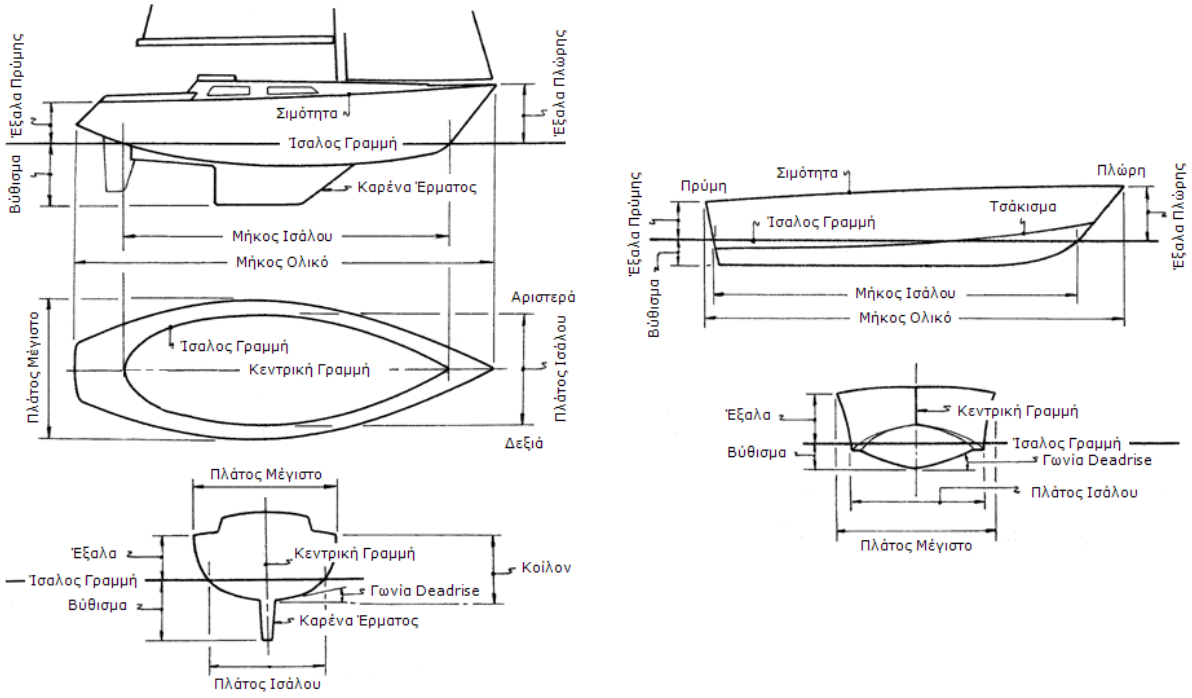
Εικόνα 3-4. Τροποποίηση του καταστρώματος για τη στήριξη της νέας θυρίδας πρόσβασης.

Εικόνα 3-5 Το ανακατασκευασμένο κατάστρωμα του σκάφους.

3.3.2 Κατασκευή εσωτερικού καταστρώματος

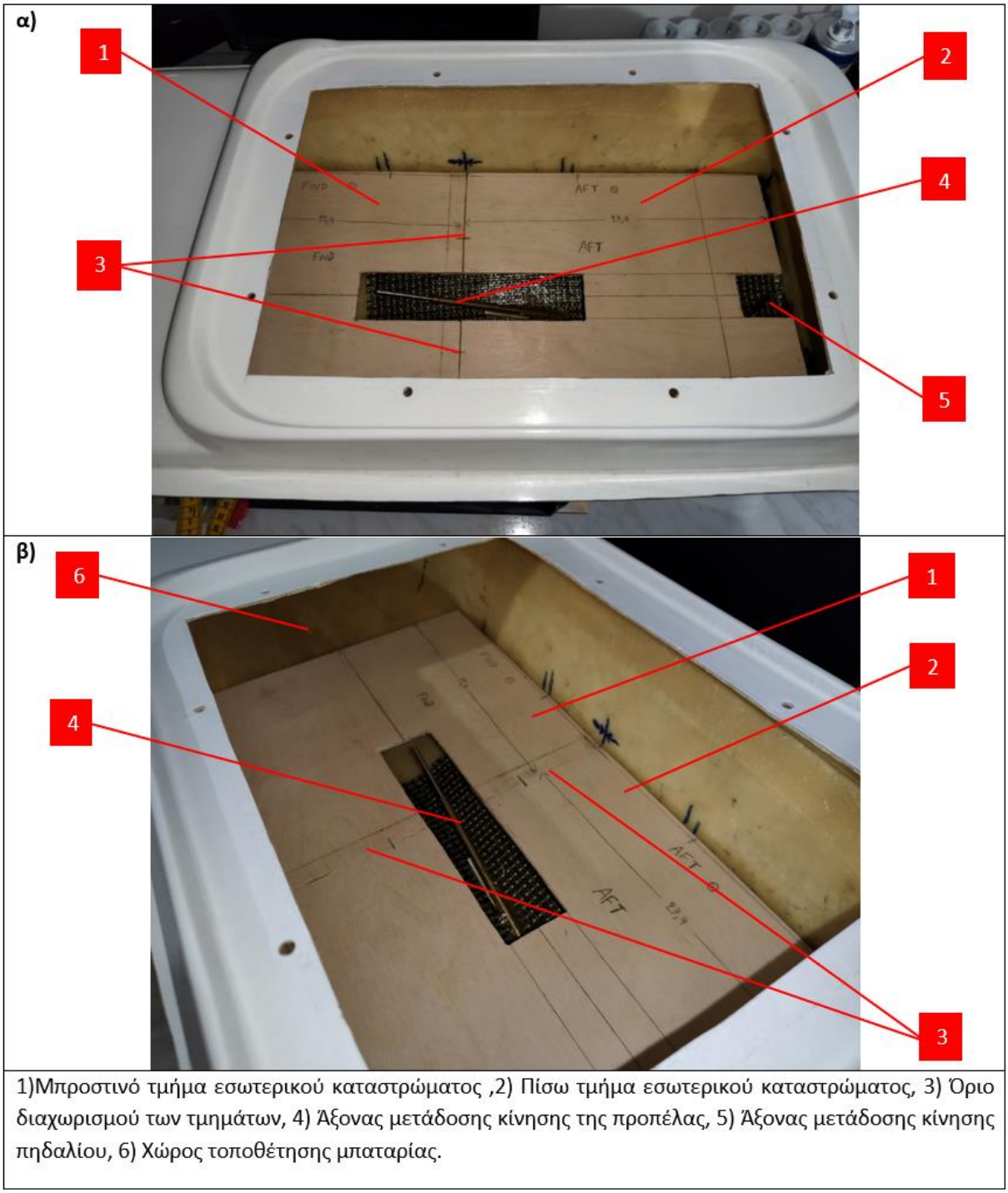
Επόμενο στάδιο της υλοποίησης είναι η δημιουργία των κατάλληλων χώρων για τη στήριξη των ηλεκτρονικών, ηλεκτρομηχανικών και μηχανικών εξαρτημάτων που θα τοποθετηθούν στο εσωτερικό του σκάφους. Η δημιουργία ενός εσωτερικού καταστρώματος με απλή γεωμετρία φάνηκε ως η καλύτερη λύση για τη συγκεκριμένη υλοποίηση. Το εσωτερικό κατάστρωμα για τη κατασκευή του, χωρίστηκε σε δύο μικρότερα ορθογώνια τμήματα τα οποία μπορούν να περάσουν μέσα από την θυρίδα πρόσβασης και να τοποθετηθούν ή να αφαιρεθούν με τη σειρά, κατά μήκος του εσωτερικού του σκάφους. Κάθε ένα από τα τμήματα αυτά κατασκευάστηκαν από σημύδα πάχους τεσσάρων χιλιοστών και διαθέτουν τις κατάλληλες εγκοπές ώστε να υπάρχει πρόσβαση στους άξονες μετάδοσης κίνησης της προπέλας αλλά και του πηδαλίου. Τα τμήματα του καταστρώματος για τη στήριξη τους, χρησιμοποιούν ως βάση το τσάκισμα που εκτείνεται στο ίδιο μήκος με την ίσαλο γραμμή και χωρίζει το βύθισμα από τα έξαλα του σκάφους όπως παρουσιάζεται στην Εικόνα 3-6 . Ως εκ τούτου, η φιλοσοφία στη κατασκευή του εσωτερικού καταστρώματος της υλοποίησης, βασίζεται στην απλότητα, τη στιβαρότητα και το χαμηλό κόστος κατασκευής. Επίσης

εξασφαλίζεται η ευκολία στην τοποθέτηση ή αφαίρεση του για μελλοντικές τροποποιήσεις που μπορεί να γίνουν στο ίδιο αλλά και στο κύτος.



Εικόνα 3-6. Απεικόνιση γενικών ναυπηγικών όρων στην ανατομία διαφόρων σκαφών [42].

Όπως φαίνεται στην εικόνα, το πρώτο κομμάτι του καταστρώματος τοποθετείται στο μπροστινό μέρος του σκάφους κοντά στη πλώρη και πάνω του θα τοποθετηθεί η βάση του κινητήρα. Μπροστά από το κατάστρωμα αυτό και πίσω από τη πλώρη, όπως απεικονίζεται στην εικόνα υπάρχει χώρος για τη τοποθέτηση της μπαταρίας. Το δεύτερο κομμάτι του καταστρώματος τοποθετείται τελευταίο στο πίσω μέρος του σκάφους προς τη πρύμνη και πάνω του υπάρχει χώρος για τη τοποθέτηση των ηλεκτρονικών συστημάτων της υλοποίησης, καθώς και του σερβομηχανισμού για τον έλεγχο του πηδαλίου. Στην εικόνα παρουσιάζεται το πίσω κομμάτι του εσωτερικού καταστρώματος.



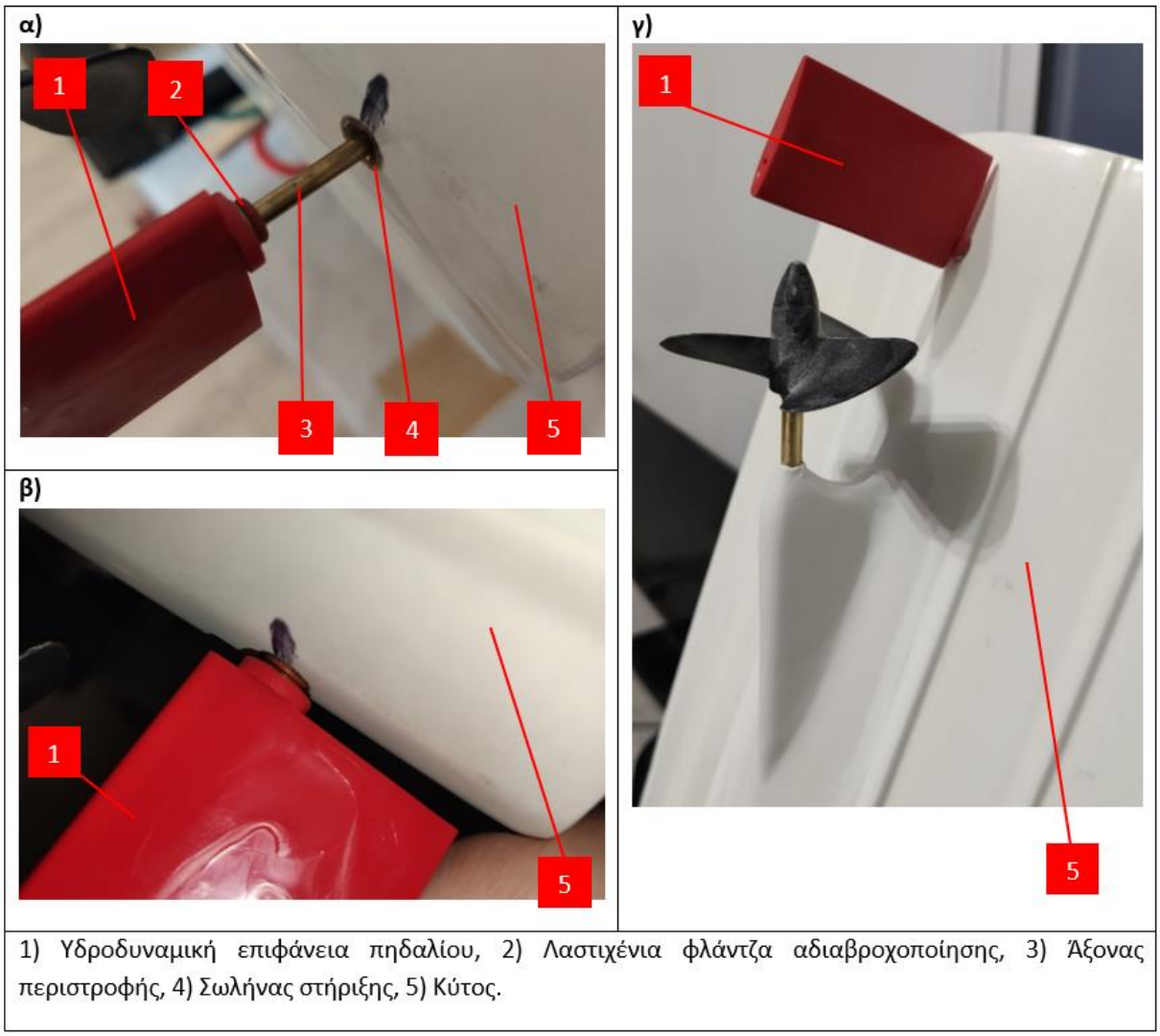
1)Μπροστινό τμήμα εσωτερικού καταστρώματος ,2) Πίσω τμήμα εσωτερικού καταστρώματος, 3) Όριο διαχωρισμού των τμημάτων, 4) Άξονας μετάδοσης κίνησης της προπέλας, 5) Άξονας μετάδοσης κίνησης πηδαλίου, 6) Χώρος τοποθέτησης μπαταρίας.

Εικόνα 3-7. Απεικόνιση και περιγραφή βασικών τμημάτων του εσωτερικού καταστρώματος.

3.3.3 Εγκατάσταση συστήματος κατεύθυνσης.

Για τον έλεγχο της πορείας του, το μη επανδρωμένο όχημα επιφανείας διαθέτει προεγκατεστημένο, ένα πλαστικό πηδάλιο μήκους εβδομήντα (70) εκατοστών και πλάτους τριών (3) εκατοστών, ενώ για το άξονα μετάδοσης της κίνησης του, διαθέτει ένα μπρούτζινο άξονα πάχους τριών (3) χιλιοστών. Ο άξονας μετάδοσης κίνησης του πηδαλίου οδηγείται στο εσωτερικό του σκάφους, μέσα από έναν μπρούτζινο σωλήνα με εσωτερική διάμετρο ίδια με

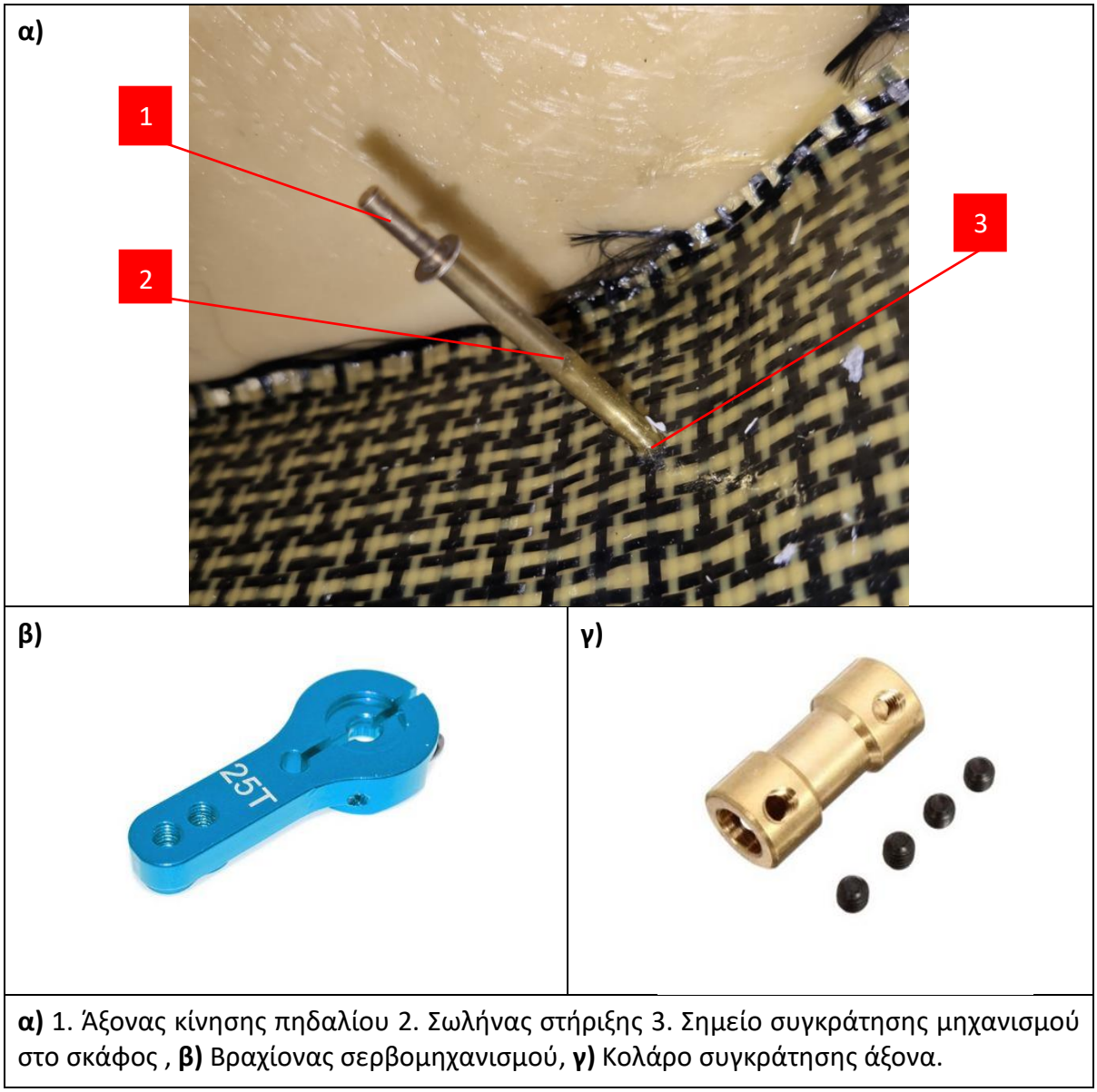
το πάχος του άξονα του πηδαλίου. Μια λαστιχένια φλάντζα και γράσο θαλάσσης που βρίσκονται ανάμεσα τον άξονα και τη σωλήνα εμποδίζουν την εισροή υδάτων στο εσωτερικό του κύτους και επιτρέπουν την ελεύθερη περιστροφή του πηδαλίου. Στην Εικόνα 3-8 παρουσιάζεται ο μηχανισμός του προεγκατεστημένου πηδαλίου όπως τοποθετείται από την εξωτερική μεριά του σκάφους.



1) Υδροδυναμική επιφάνεια πηδαλίου, 2) Λαστιχένια φλάντζα αδιαβροχοποίησης, 3) Άξονας περιστροφής, 4) Σωλήνας στήριξης, 5) Κύτος.

Εικόνα 3-8. Προεγκατεστημένος μηχανισμός πηδαλίου, όπως φαίνεται από την εξωτερική μεριά του σκάφους.

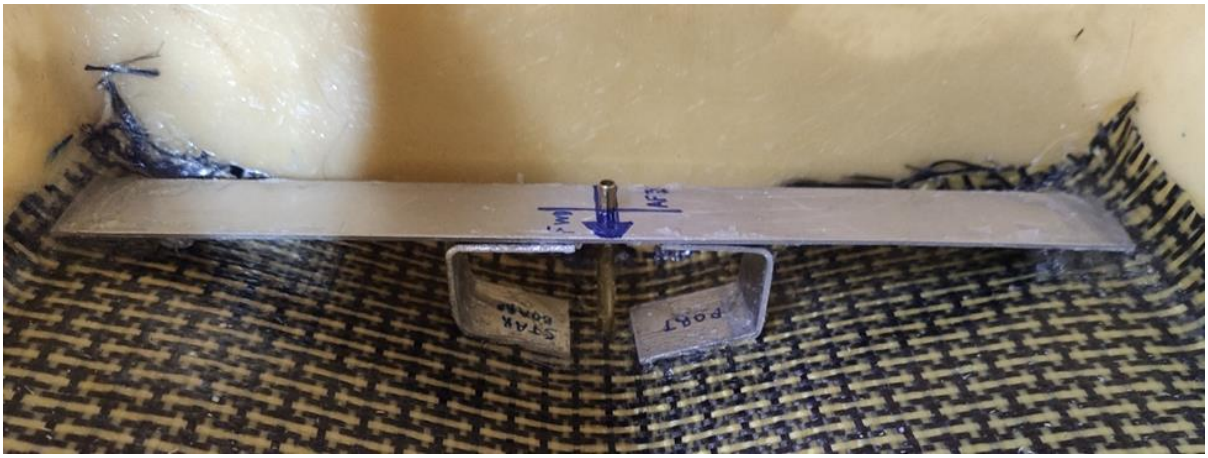
Στο εσωτερικό του οχήματος, η κίνηση του πηδαλίου μεταδίδεται από τον άξονα σε ένα βραχίονα σερβομηχανισμού που βιδώνεται πάνω του. Επιπλέον, εσωτερικά της κατασκευής, ένα μεταλλικό κολάρο σφίγγεται πάνω στον άξονα συγκρατώντας το πηδάλιο στη θέση του και εμποδίζοντας το πηδάλιο να ολισθήσει προς τα κάτω λόγω της βαρύτητας, γεγονός που θα το αποσπούσε από το όχημα. Στην Εικόνα 3-9 απεικονίζονται ο άξονας του πηδαλίου που οδηγείται στο εσωτερικό του κύτους , το κολάρο συγκράτησης και ο βραχίονας σερβομηχανισμού που χρησιμοποιήθηκαν.



α) 1. Άξονας κίνησης πηδαλίου 2. Σωλήνας στήριξης 3. Σημείο συγκράτησης μηχανισμού στο σκάφος , **β)** Βραχίονας σερβομηχανισμού, **γ)** Κολάρο συγκράτησης άξονα.

Εικόνα 3-9. Τα εξαρτήματα του άξονα μετάδοσης κίνησης του πηδαλίου.

Παρατηρήθηκε από πρώτες δοκιμές στο νερό, ότι δημιουργούνταν σοβαρές στρεβλώσεις του μηχανισμού περιστροφής του πηδαλίου λόγω της αντίστασης του νερού. Αυτό οφείλονταν στη μικρή διάμετρο του άξονα και του σωλήνα μετάδοσης κίνησης του πηδαλίου, το οποίο στηρίζονταν μόνο σε ένα σημείο, αυτό του τοιχώματος του κύτους (σημείο 3, Εικόνα 3-9α). Επομένως ως λύση επιλέχθηκε η ενίσχυση της κατασκευής με τη δημιουργία μιας αλουμινένιας πλατφόρμας σταθεροποίησης του σωλήνα μετάδοσης κίνησης, όπως φαίνεται στην Εικόνα 3-10. Η κατασκευή κολλήθηκε στο εσωτερικό του κύτους με σιλικόνη υψηλής αντοχής.



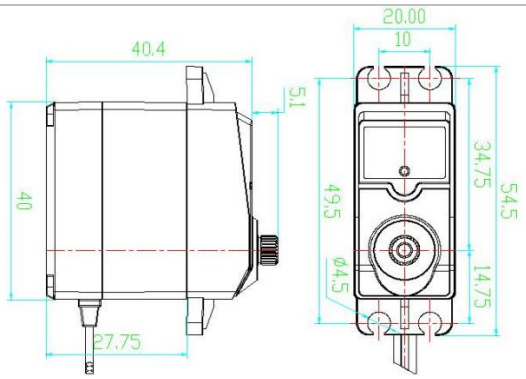
Εικόνα 3-10. Η πλατφόρμα ενίσχυσης του πηδαλίου.

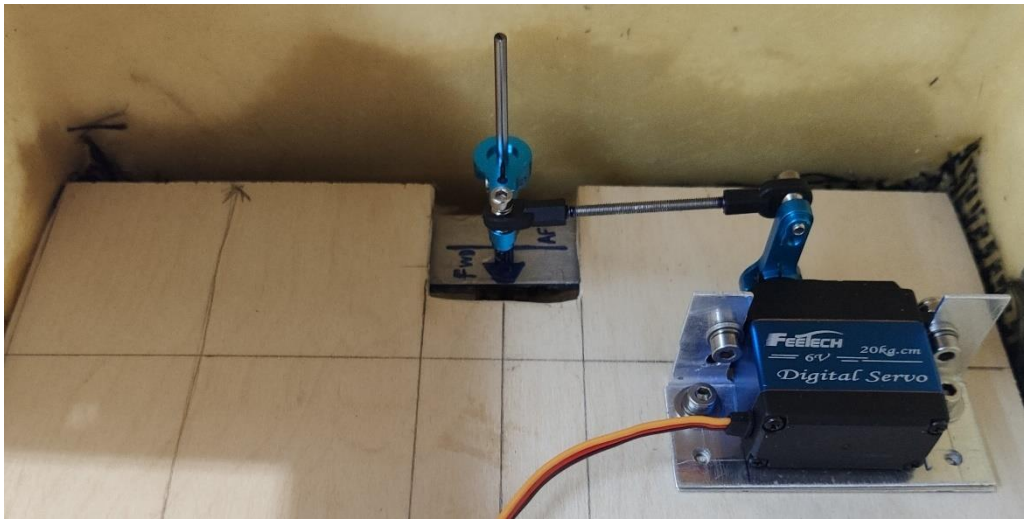
Έχοντας εξασφαλίσει πλέον τη στιβαρότητα του μηχανισμού περιστροφής του πηδαλίου, τα εξαρτήματα στην Εικόνα 3-9 που περιεγράφηκαν προηγουμένως, επανατοποθετήθηκαν μαζί με μία μπιλιοφόρο δοκό μετάδοσης κίνησης προς το βραχίονα του σερβοκινητήρα. Ο Σερβοκινητήρας αντίστοιχα στηρίζεται οριζόντια στο πίσω εσωτερικό κατάστρωμα και ο άξονας περιστροφής του, είναι κάθετος στον άξονα περιστροφής του πηδαλίου. Η απλουστευμένη αυτή, γεωμετρική διάταξη του σερβομηχανισμού, δεν θα ήταν εφικτή χωρίς την μπιλιοφόρο δοκό που επιτρέπει τη μετάδοση κίνησης, μεταξύ δύο αξόνων περιστροφής που έχουν διαφορετική γωνία μεταξύ τους. Στην Εικόνα 3-11 και Εικόνα 3-12 παρουσιάζεται η διάταξη μετάδοσης κίνησης από το σερβοκινητήρα στο πηδάλιο. Ο ενεργοποιοιητής που επιλέχθηκε για τη συγκεκριμένη υλοποίηση έχει τα εξής χαρακτηριστικά:

Πίνακας 3-1. Χαρακτηριστικά ενεργοποιοιητή. [5]

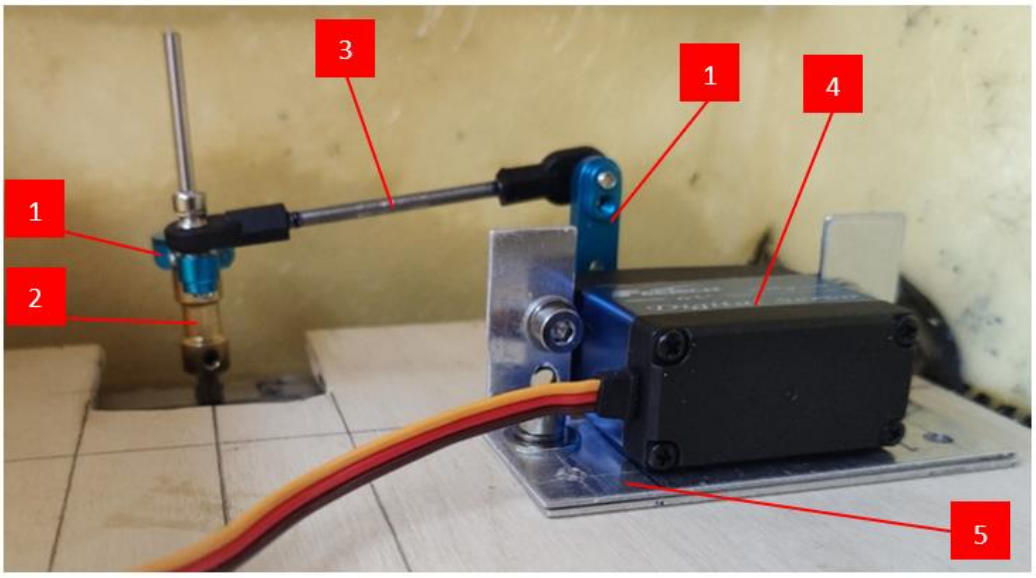
General Information	**Characteristics per Operating Voltage Range**	
Manufacturer: Feetech	**Operating Voltage Range: 4.8V**	**Operating Voltage Range: 6V**
Model : FS5323M	Idle current(at stopped): 5mA	Idle current(at stopped): 7mA
Servo Standard : 25T (Futaba)	No load speed: 0.22sec/60°	No load speed: 0.16sec/60°
Gear Type: Metal	Running current(at no load): 150 mA	Runnig current(at no load): 180 mA
Rotation: 180°	Peak stall torque: 19kg.cm / 263.8oz.in	Peak stall torque: 21.5kg.cm / 298oz.in
Servo Operation: Digital	Stall current: 1200 mA	Stall current: 1200 mA
Servo Size : Standard		
Command signal: Pulse width modulation		

[5] Servo Datasheet: https://grobotronics.com/images/companies/1/datasheets/File-1525770326.pdf?1526982539123

Pulse width range: 500~2500μsec	
PWM Neutral position: 1500 μsec	
PWM Dead band width: 4 μsec	
Dimensions: 40 x 20 x 40.4 mm	
Weight: 67g	
Horn Dimensions: 35 x 15 x 6 mm	



Εικόνα 3-11. Η διάταξη του σερβομηχανισμού.



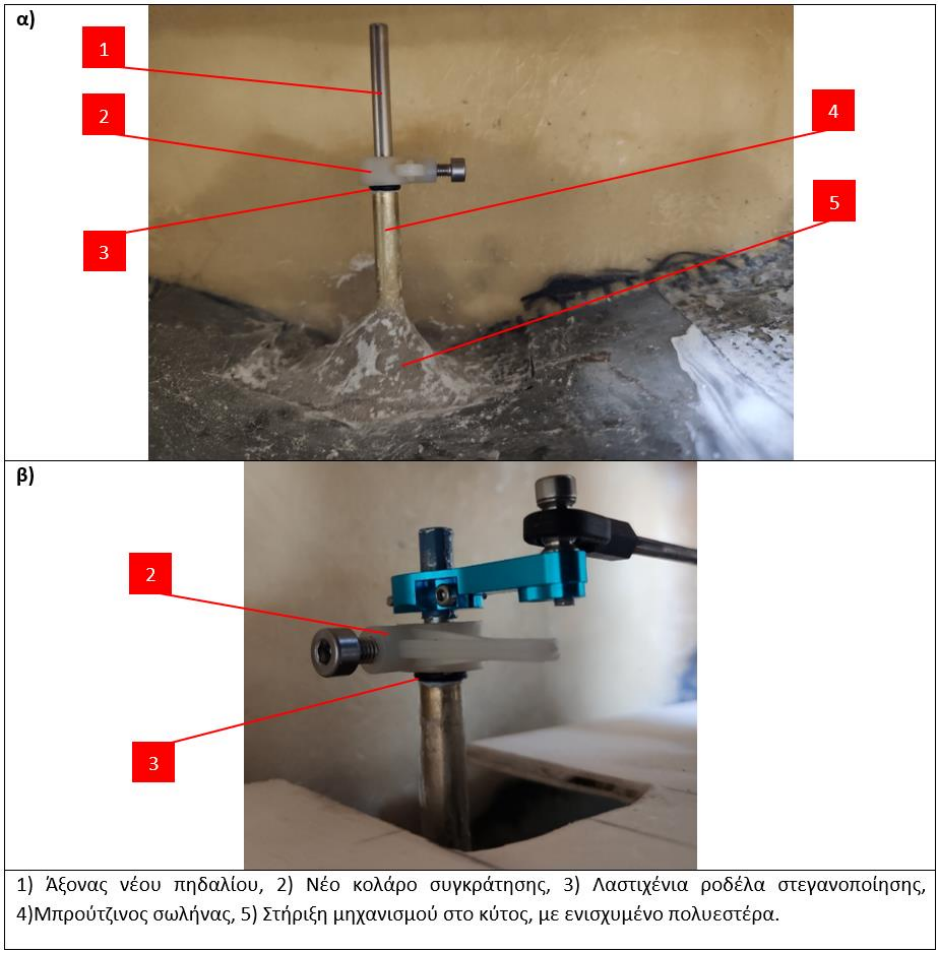
1) Βραχίονας σερβομηχανισμού, 2) Κολάρο συγκράτησης αξόνων 3) Μπιλιοφόρος δοκός, 4) Ενεργοποιητής σερβό, 5) Βάση στήριξης καταστρώματος.

Εικόνα 3-12. Κοντινό πλάνο διάταξης σερβομηχανισμού.

Ύστερα από ένα μικρό κύκλο δοκιμών με τον μηχανισμό κατεύθυνσης σε άμεσο έλεγχο από τον χειριστή μέσω τηλεχειρισμού, παρατηρήθηκε ότι το σκάφος σε κάθε αλλαγή πορείας ταλαντώνονταν δεξιά και αριστερά. Ταυτόχρονα σε ελιγμούς ανάστροφης ώθησης το σκάφος δεν ανταποκρίνονταν αποτελεσματικά στην εντολή του χειριστή για αλλαγή κατεύθυνσης. Τα προβλήματα που αναφέρθηκαν, οφείλονται στα παρακάτω αίτια:

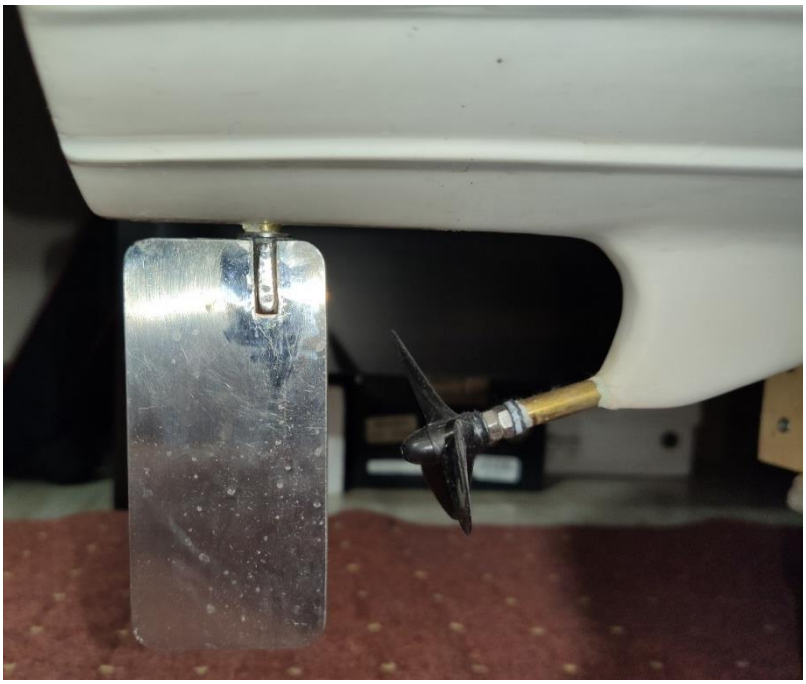
1. Ο μεταλλικός άξονας περιστροφής και το πτερύγιο του πηδαλίου ήταν πολύ λεπτά με αποτέλεσμα να στρεβλώνονται και να ταλαντώνονται από τις δυνάμεις που ασκούσε πάνω τους το νερό.
2. Η πρύμνη του μοντέλου σκάφους που επιλέχθηκε, έχει πολύ χαμηλή υδροδυναμική απόδοση σε ανάστροφη πορεία του οχήματος, με αποτέλεσμα να δημιουργείται το φαινόμενο της μετατόπισης του σημείου περιστροφής (Pivot Point) [43].

Για να επιλυθεί το πρόβλημα των ταλαντώσεων της πορείας του οχήματος, το προεγκατεστημένο πηδάλιο αφαιρέθηκε και τοποθετήθηκε ένα νέο, μεγαλύτερων διαστάσεων από ανοξείδωτο ατσάλι. Ο άξονας μετάδοσης κίνησης είναι πλέον διαμέτρου 6 χιλιοστών και η στήριξη του αντίστοιχου σωλήνα έγινε εξολοκλήρου με ενισχυμένο πολυεστέρα επάνω στο κύτος. Επιπλέον το πτερύγιο επεκτάθηκε σε διαστάσεις με μήκος 10 και πλάτος 4.5 εκατοστών αντίστοιχα. Μετά από πολλές δοκιμές ο νέος ενισχυμένος μηχανισμός κατεύθυνσης απέδωσε εξαιρετικά, με αποτέλεσμα τον άμεσο και ακριβή έλεγχο του σκάφους από το χειριστή. Ταυτόχρονα βελτιώθηκε αισθητά ο έλεγχος σε ανάστροφη πορεία με χαμηλές ταχύτητες. Στην Εικόνα 3-13 και Εικόνα 3-14 φαίνεται το νέο πηδάλιο μέσα κι έξω από το σκάφος.



1) Άξονας νέου πηδαλίου, 2) Νέο κολάρο συγκράτησης, 3) Λαστιχένια ροδέλα στεγανοποίησης, 4)Μπρούτζινος σωλήνας, 5) Στήριξη μηχανισμού στο κύτος, με ενισχυμένο πολυεστέρα.

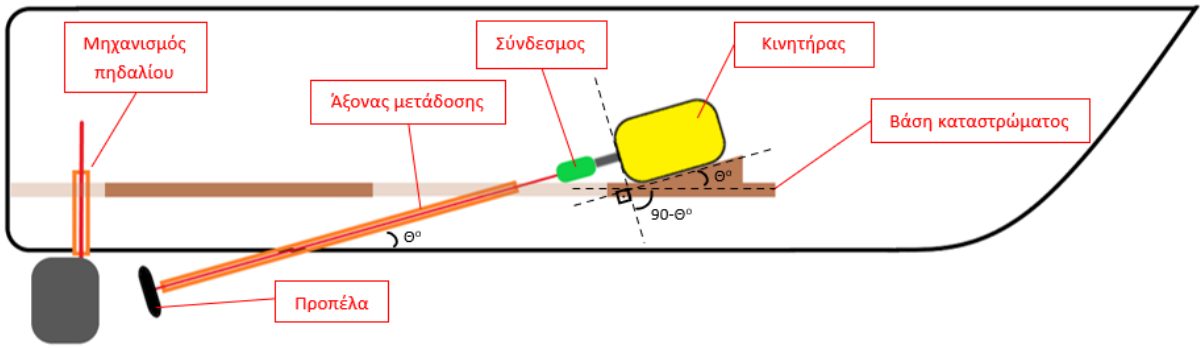
Εικόνα 3-13. Η εγκατάσταση του νέου πηδαλίου.



Εικόνα 3-14. Εξωτερική όψη.

3.3.4 Εγκατάσταση συστήματος προώθησης.

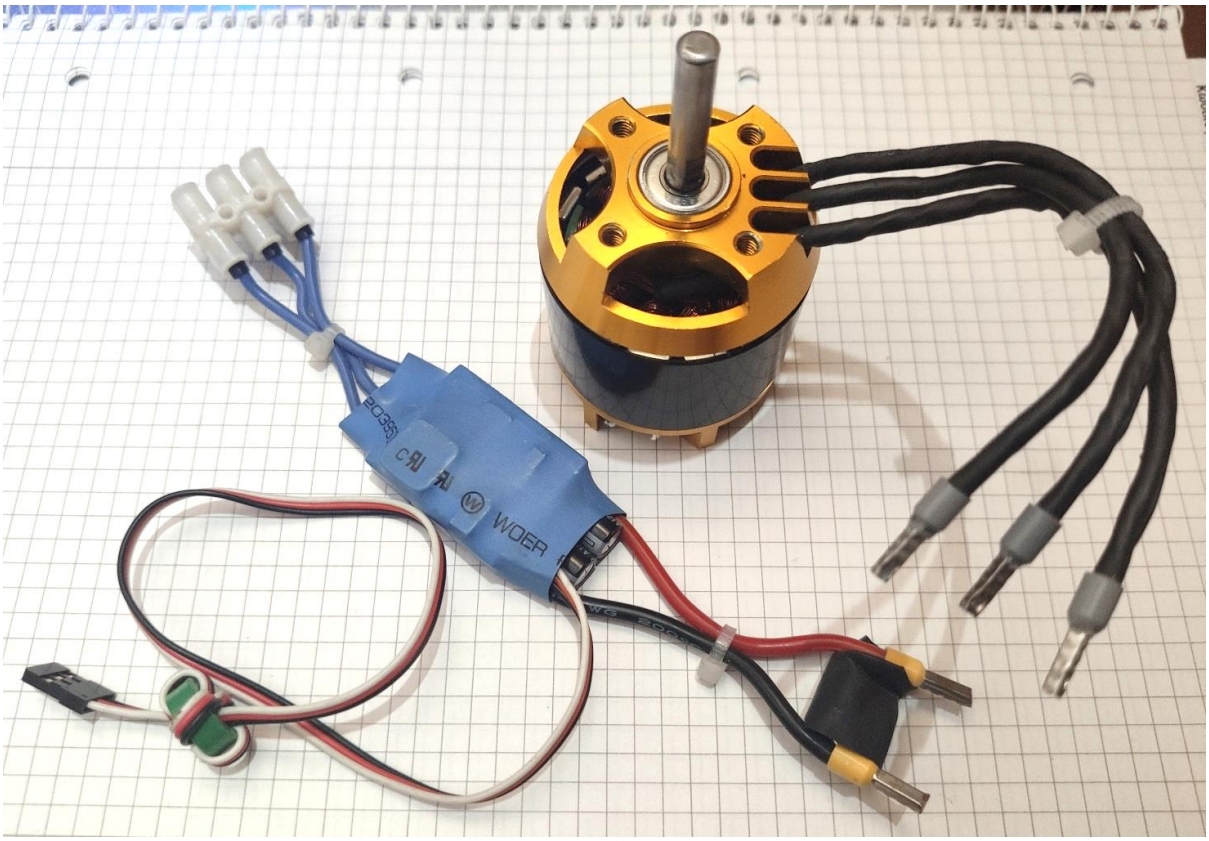
Το μοντέλο του σκάφους για τη προώθηση του, διαθέτει ήδη εγκατεστημένο ένα σύστημα μετάδοσης μονής προπέλας και μηχανισμό άξονα παρόμοιο σε λειτουργία με αυτόν του συστήματος κατεύθυνσης. Η διαφορά είναι ότι η μετάδοση έχει μήκος 30 εκατοστά, διάμετρο 4 χιλιοστά και υλικό κατασκευής από ανοξείδωτο ατσάλι. Ο άξονας προεξέχει από τη καρίνα του σκάφους υπό γωνία (βλέπε Εικόνα 3-14), έτσι ώστε να υπάρχει αρκετός χώρος στο εσωτερικό για να στηριχθεί ο κινητήρας και να συνδεθεί με το σύστημα μετάδοσης, σύμφωνα με την Εικόνα 3-15. Όπως είχε αναφερθεί στην ενότητα 3.2, για την κίνηση του οχήματος θα χρησιμοποιηθεί ένα επαγωγικό τριφασικό μοτέρ συνεχούς ρεύματος (DC) που δίνει το πλεονέκτημα υψηλής απόδοσης και ελάχιστων φθορών. Για την οδήγηση του μοτέρ, είναι απαραίτητη η χρήση ενός αντιστροφέα (inverter) με μεταβλητή συχνότητα εναλλαγής των τριών φάσεων, για τον έλεγχο της ταχύτητας του κινητήρα. Επιπλέον το κύκλωμα οδήγησης αυτού του τύπου ηλεκτροκινητήρα είναι γνωστό και ως ESC (Electronic Speed Control).



Εικόνα 3-15. Διατομή προφίλ εσωτερικού.

Τα ηλεκτρονικά μέρη του συστήματος προώθησης παρουσιάζονται στην Εικόνα 3-16. Η συγκεκριμένη ενότητα όμως, εστιάζεται:

- στα χαρακτηριστικά του κινητήρα,
- το τρόπο στήριξης του στο εσωτερικό του σκάφους
- και της σύνδεσης του με τον άξονα της προπέλας.



Εικόνα 3-16. Ο κινητήρας και το ESC.

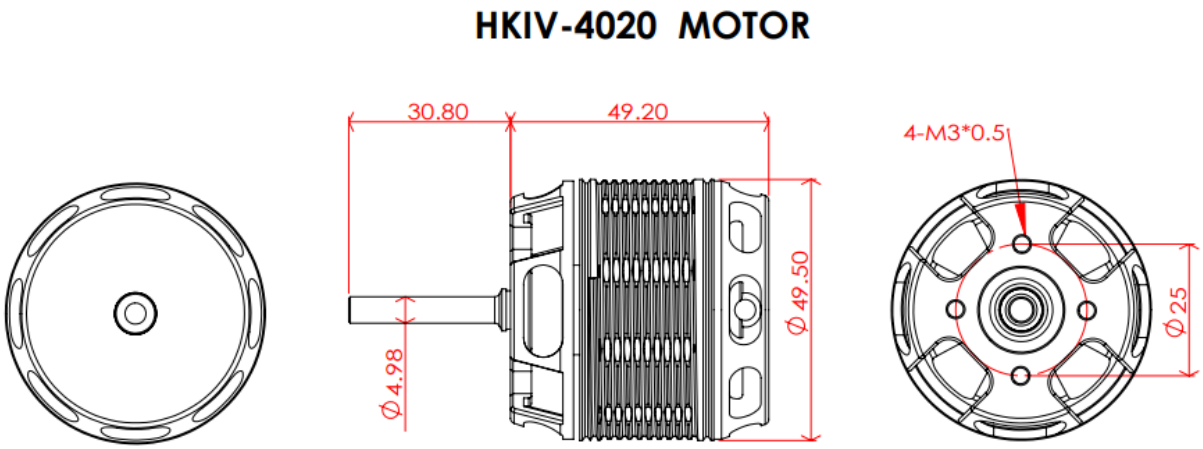
Το μοτέρ που επιλέχθηκε αποτελεί μια παλιά έκδοση του μοντέλου HKIV-4020-1320KV της εταιρείας Scorpion. Παρόλα αυτά τα τεχνικά χαρακτηριστικά του, παραμένουν ίδια με την νέα του έκδοση. Στον Πίνακα 3-2 και στην Εικόνα 3-17 παρουσιάζονται τα ηλεκτρονικά και μηχανολογικά χαρακτηριστικά του κινητήρα.

Πίνακας 3-2. Τεχνικά χαρακτηριστικά ηλεκτροκινητήρα.[6]

Manufacturer	Scorpion
Model	HKIV-4020-1320KV
Stator Diameter	40 mm(1.57 in)
Stator Thickness	20 mm(0.79 in)
No. Of Stator Arms	12
Magnet Poles	10
Motor Wind	3+4 Turn Delta

[6] Motor Datasheet: https://www.scorpionsystem.com/catalog/helicopter/motors_4/hkiv-40/HKIV_4020_1320/

Motor Wire	1-Strand 1.7mm
Motor Kv	1320KV RPM/Volt
No-Load Current (IO/10v)	5.74 Amps
Motor Resistance (RM)	0.006 Ohms
Max Continuous Current	90 Amps
Max Continuous Power	1998 Watts
Weight	310 Grams (10.94 oz)
Outside Diameter	49.50 mm (1.95 in)
Shaft Diameter	4.98 mm (0.20 in)
Body Length	49.2 mm (1.94 in)
Overall Shaft Length	80 mm (3.15 in)
Max Lipo Cell	6s
Peak Continuous Current	105 Amps (5 seconds)
Peak Continuous Power	2331 Watts (5 seconds)
Motor Timing	5deg
Drive Frequency	8kHz



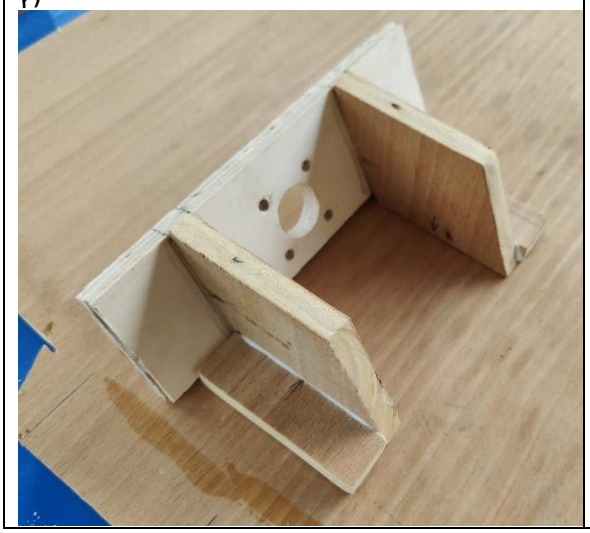
Εικόνα 3-17. Μηχανολογικό σχέδιο μοτέρ.

Χρησιμοποιώντας το μηχανολογικό σχέδιο του ηλεκτροκινητήρα (Εικόνα 3-17) και έχοντας σαν οδηγό το σχέδιο στην Εικόνα 3-15 σχεδιάστηκε το σκαρίφημα μίας πρωτότυπης βάσης στήριξης που θα προσαρτηθεί στο κατάστρωμα. Η βάση κατασκευάστηκε από ξύλο κόντρα πλακέ θαλάσσης πάχους 8 χιλιοστών και ενισχύθηκε δομικά με ανοξείδωτες βίδες. Η κλίση στήριξης του κινητήρα μετρήθηκε με μοιρογνωμόνιο. Γνωρίζοντας ότι το κατάστρωμα είναι παράλληλο στη διαμήκη γραμμή της καρίνας του σκάφους, τότε η γωνία κλίσης αυτής της γραμμής με τον άξονα μετάδοσης κίνησης της προπέλας, είναι ίδια με την γωνία που δημιουργεί ο άξονας περιστροφής του κινητήρα με το κατάστρωμα. Αυτό το συμπέρασμα φαίνεται ξανά στο σχέδιο στην Εικόνα 3-15 όπου Θ είναι η γωνία αναφοράς. Με τις πληροφορίες που αναφέρθηκαν κατασκευάστηκε το στήριγμα του μοτέρ όπως φαίνεται στην Εικόνα 3-18.

α)

β)

γ)



δ)

Εικόνα 3-18. Η βάση στήριξης του κινητήρα.

Η ολοκληρωμένη βάση στηρίχθηκε στο εσωτερικό μπροστινό κατάστρωμα του σκάφους και ευθυγραμμίστηκε όσο το δυνατόν καλύτερα με τον άξονα της προπέλας. Ένας σωλήνας ίδιας διαμέτρου στις άκρες του, με τους αντίστοιχους άξονες προπέλας και μοτέρ, βοήθησε σε αυτή τη διεργασία ευθυγράμμισης.

Το επόμενο βήμα, ήταν η επιλογή του κατάλληλου μηχανικού συνδέσμου μεταξύ ηλεκτροκινητήρα και προπέλας. Ο σύνδεσμος θα πρέπει να έχει υψηλή αντοχή στην ροπή του κινητήρα και σε καταπονήσεις που μπορεί να προκληθούν από μικρές αλλαγές στην κατά μήκος απόσταση, μεταξύ των δύο συζευγμένων αξόνων περιστροφής. Επίσης ο σύνδεσμος θα πρέπει να μεταδώσει κίνηση μεταξύ δύο αξόνων που μπορεί να μην είναι ευθυγραμμισμένοι μεταξύ τους. Το εξάρτημα αυτό πρέπει να έχει απλό σχεδιασμό για να ελαττωθούν τα σημεία φθοράς κατά τη λειτουργία του. Για αυτό το σκοπό επιλέχθηκε ένας σύνδεσμος «Universal Joint» τύπου «Dogbone» ή αλλιώς γνωστός ως «pinned ball joint», ο οποίος επιτρέπει την μετάδοση κίνησης σε ένα ευρύ κώνο κλίσης, ενώ ταυτόχρονα παρέχει ανοχή στην μεταβολή απόστασης σύζευξης. Στην Εικόνα 3-19 παρουσιάζεται η βάση με το μοτέρ τοποθετημένη στο εσωτερικό του σκάφους και ο σύνδεσμος που τοποθετήθηκε μεταξύ κινητήρα και προπέλας.

α)

β)

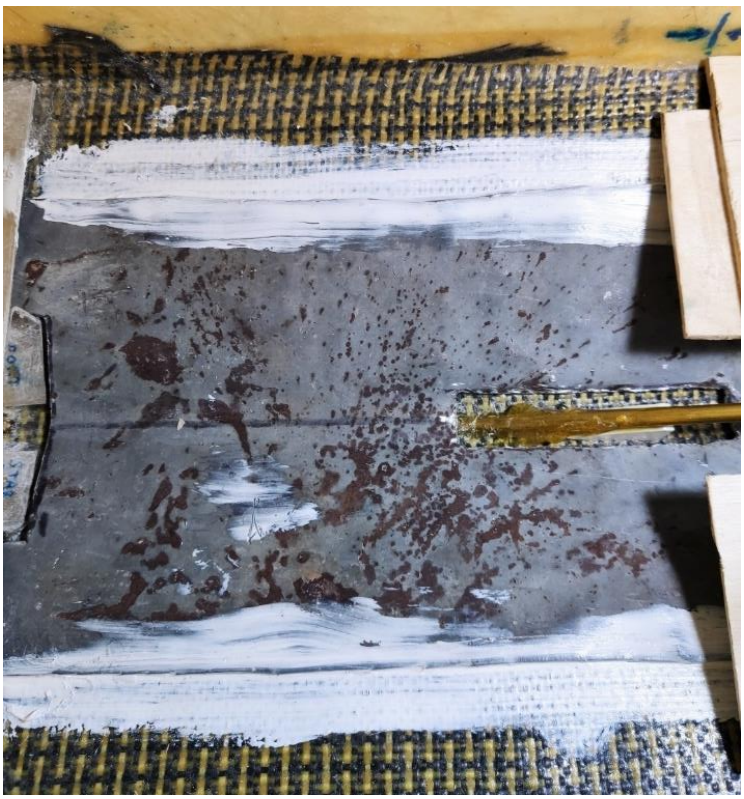
Εικόνα 3-19. Η βάση του κινητήρα και ο σύνδεσμος μετάδοσης.

3.3.5 Λοιπές μηχανολογικές τροποποιήσεις σκάφους

Παρατηρήθηκε ότι η προσθήκη ή/και τροποποίηση εξαρτημάτων στο σκάφος όπως:

- το εξωτερικό και εσωτερικό κατάστρωμα,
- ο μηχανισμός προώθησης και κατεύθυνσης,
- η μπαταρία και τα υπόλοιπα ηλεκτρονικά,

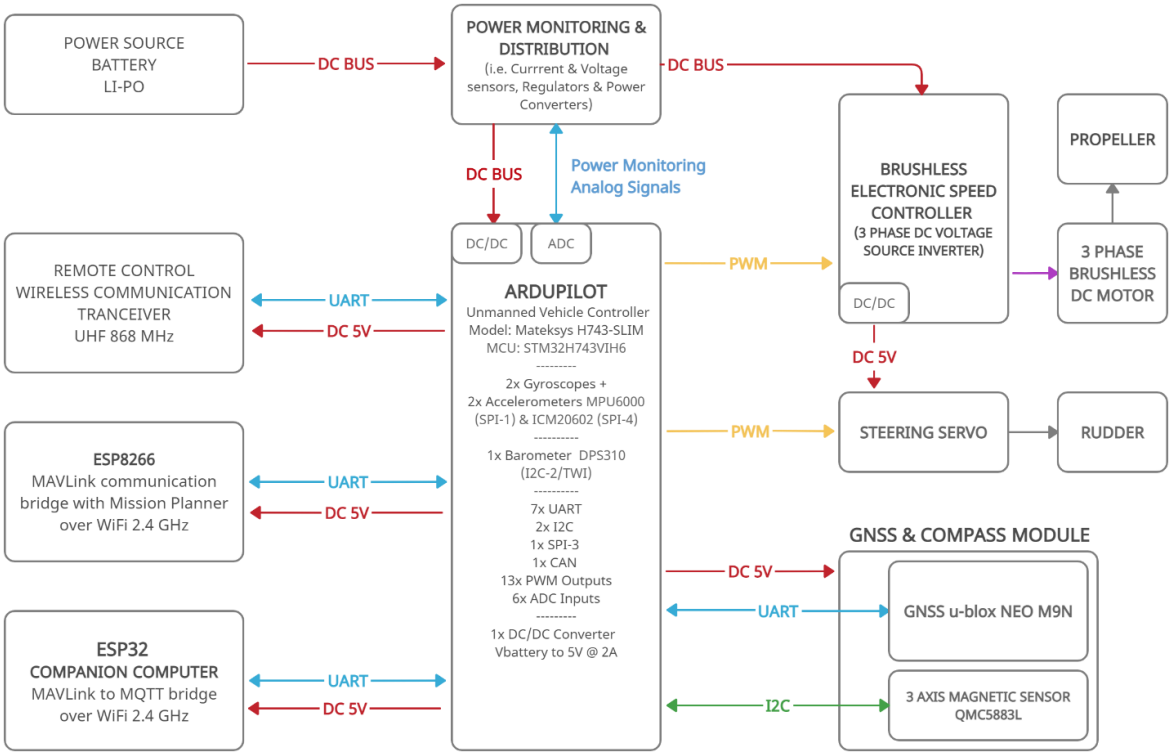
μετατόπισε το κέντρο βάρους του σκάφους σε μεγαλύτερο ύψος. Αυτό έχει ως αποτέλεσμα να αυξηθεί το «Rolling» ή αλλιώς η «Διατοίχιση» του σκάφους. Για να επιλυθεί το πρόβλημα αυτό, τοποθετήθηκε έρμα στη καρίνα του σκάφους. Το υλικό που χρησιμοποιήθηκε είναι φύλλο από μόλυβδο που χρησιμοποιείται ως οικιακό μονωτικό. Επιλέχθηκε κυρίως λόγω του μεγάλου βάρους σε αναλογία όγκου, της ευκολίας που έχει να κοπεί στις επιθυμητές διαστάσεις και της ευκολίας να πλαστεί στο επιθυμητό σχήμα. Στην Εικόνα 3-20, απεικονίζεται το μολύβδινο φύλλο έρματος που κολλήθηκε με σιλικόνη στη καρίνα του κύτους, ανάμεσα στους άξονες του πηδαλίου και της προπέλας.



Εικόνα 3-20. Έρμα σκάφους στη καρίνα.

3.4 Ηλεκτρονικό μέρος

Η ενότητα αυτή περιλαμβάνει τη περιγραφή των τεχνικών χαρακτηριστικών και τις διαδικασίες εγκατάστασης όλων των ηλεκτρονικών εξαρτημάτων που τοποθετήθηκαν στο σκάφος. Επίσης περιγράφονται οι παραμετροποιήσεις στο λογισμικό των ηλεκτρονικών συστημάτων όπου αυτό απαιτείται. Στην Εικόνα 3-21 παρουσιάζεται γραφικά η αρχιτεκτονική που ακολουθήθηκε για την υλοποίηση του ηλεκτρονικού μέρους του οχήματος.



Εικόνα 3-21. Αρχιτεκτονική μη επανδρωμένου σκάφους επιφανείας.

3.4.1 Η μπαταρία

Για το όχημα χρειάζεται μια μπαταρία με υψηλή χωρητικότητα, μικρές διαστάσεις και χαμηλό βάρος. Επίσης, πέρα από την μεγάλη ενεργειακή πυκνότητα ο συσσωρευτής θα πρέπει να μπορεί να τροφοδοτήσει φορτία με υψηλές απαιτήσεις σε ρεύμα. Ο κινητήρας με το κύκλωμα οδήγησης του (ESC) συνδυαστικά απαιτούν συνεχή τροφοδοσία δεκάδων Ampere απευθείας από τη μπαταρία (40 Ampere μέγιστο συνεχές ρεύμα). Τα υπόλοιπα ηλεκτρονικά εξαρτήματα του οχήματος συμπεριλαμβανομένου και του ενεργοποιητή (Servo) του πηδαλίου, δεν ξεπερνούν συνολικά τα 2.5 Ampere με τάση τροφοδοσίας 5 Volt. Σύμφωνα με τις ανάγκες που αναφέρθηκαν, η καλύτερη τεχνολογία μπαταρίας για το όχημα αυτό είναι τα πολυμερή λιθίου (LiPo). Για τις ανάγκες της υλοποίησης υπήρχε ήδη διαθέσιμη μια μπαταρία LiPo με τα ακόλουθα τεχνικά χαρακτηριστικά του Πίνακα 3-3.

Πίνακας 3-3. Τεχνικά χαρακτηριστικά μπαταρίας. [7]

Configuration	4S1P (4 Cells in series, 1 package)
Connectors	1 x XT90 for power leads , 1 x 5pins JST XH for balance charging
Battery Nominal Voltage	4S-14,8 Volt
Nominal Voltage per Cell	3.7 Volt
End-of-charge voltage per Cell	4.2 Volt
Capacity	5400 mAh
Discharge rate (Continuous)	50 C (50 C x 5.4 Ah = 270 Ampere)
Charge Rate	1 C (1C x 5.4Ah = 5.4 Ampere to fully charge in 1 hour)
Weight	0.535 Kg

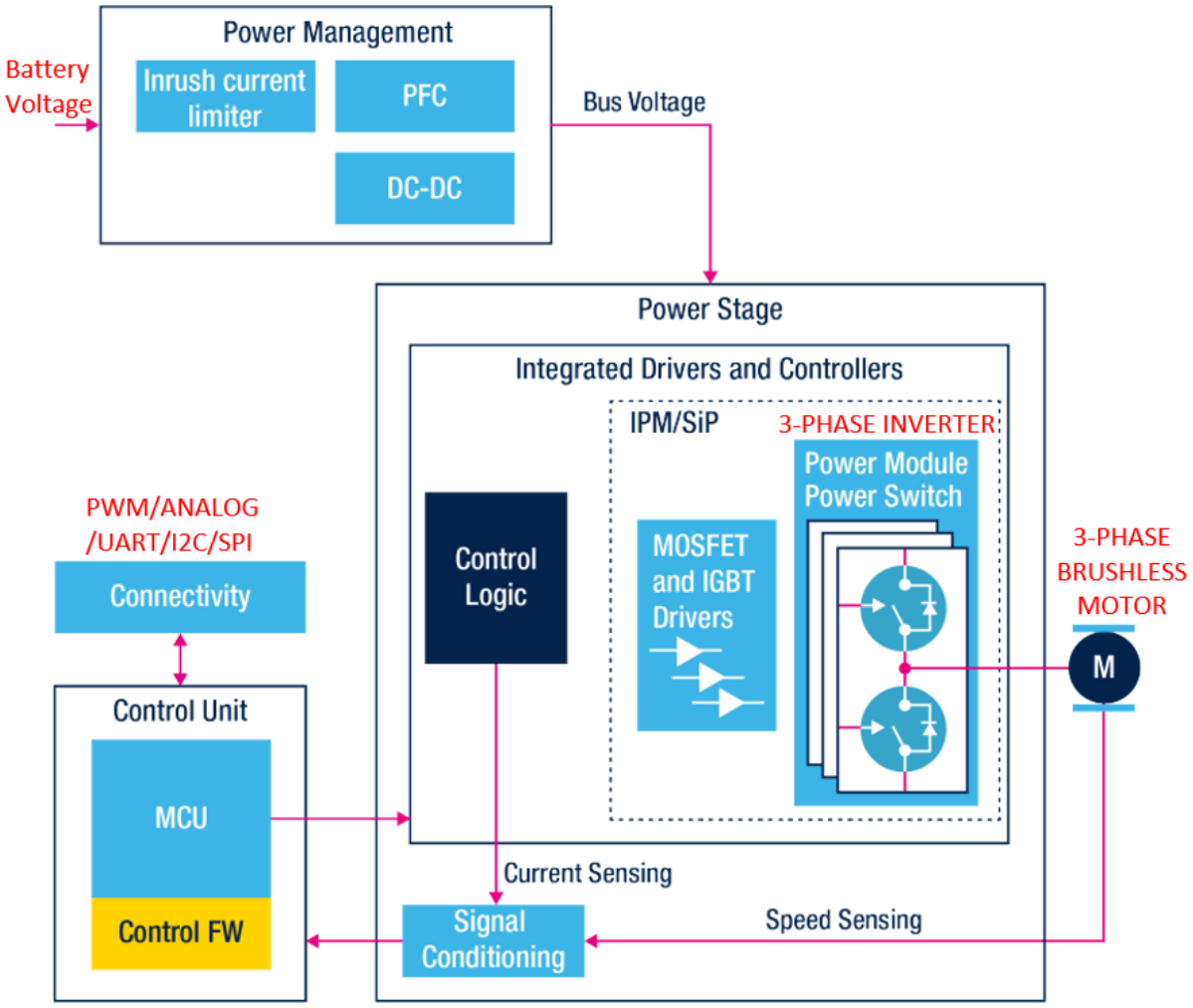
Εικόνα 3-22. Η μπαταρία του οχήματος.

3.4.2 Το κύκλωμα ελέγχου ταχύτητας κινητήρα ESC

Όπως αναφέρθηκε στην υποενότητα 3.3.4, το ESC (Electronic Speed Control) για επαγωγικούς κινητήρες, όπως αυτός της συγκεκριμένης υλοποίησης, είναι ουσιαστικά μια ελεγχόμενη γέφυρα τριών φάσεων. Η διάταξη αποτελείται από διακοπτικά στοιχεία ημιαγωγών FET, που μετατρέπουν τη τροφοδοσία μιας συνεχούς πηγής (DC) σε τρείς κατάλληλα εναλλασσόμενες φάσεις, για τον έλεγχο ενός επαγωγικού τριφασικού κινητήρα. Οι φάσεις διαμορφώνονται κατάλληλα από το κύκλωμα οδήγησης της γέφυρας του ESC για την επιτάχυνση επιβράδυνση αλλά και την αλλαγή της φοράς στρέψης του κινητήρα. Επομένως αυξάνοντας την ταχύτητα εναλλαγής φάσεων, το μοτέρ μπορεί να επιταχύνει και το αντίστροφο, ενώ με την εναλλαγή της αλληλουχίας μεταξύ δύο εκ των τριών φάσεων μπορεί να αντιστραφεί η κίνηση του. Το

[7] Battery Specs: https://www.racerc.gr/products/brainergy-lipo-battery-4s1p-14-8v-5400mah-45c-xt90-hardcase

σήμα ελέγχου του ESC μπορεί να είναι ένας παλμός διαμορφωμένος κατά πλάτος PWM, κοινός σε χαρακτηριστικά με αυτούς που χρησιμοποιούνται στους ενεργοποιητές Servo, ενώ σε πιο ανεπτυγμένα συστήματα μπορεί να είναι ψηφιακά δεδομένα μέσω σειριακού πρωτοκόλλου UART [44] [45].



Εικόνα 3-23. Γενικό διάγραμμα κυκλώματος του ESC επαγωγικού κινητήρα [46].

Σύμφωνα με τα τεχνικά χαρακτηριστικά του κινητήρα που αναλύθηκαν στην υποενότητα 3.3.4, απαιτείται ένα ESC που θα μπορεί να παρέχει συνεχή παροχή ρεύματος έως 90 Ampere με μέγιστη τάση τροφοδοσίας τα 6S δηλαδή 22.2 Volt. Η πλήρης ισχύς, δηλαδή 90A x 22.2V = 1998 Watt που μπορεί να αποδώσει το μοτέρ για το όχημα της υλοποίησης, είναι πολύ μεγάλη. Ως εκ τούτου, ο κίνδυνος για ζημιές στα μηχανικά μέρη του συγκεκριμένου συστήματος πρόωσης, είναι αρκετά υψηλός. Επίσης για την επιλογή καινούριας μπαταρίας και κυκλώματος ESC αντίστοιχα, το κόστος θα ανέβαινε εκθετικά. Έτσι στα πλαίσια της υλοποίησης αυτής, επιλέχθηκε ένα παλαιό μεταχειρισμένο κύκλωμα ελέγχου ταχύτητας με μέγιστη τάση τροφοδοσίας τα 22.2 Volt και μέγιστο συνεχές ρεύμα τροφοδοσίας τα 40 Ampere. Η υποστήριξη χαμηλότερου ρεύματος τροφοδοσίας της γέφυρας που ενσωματώνει αυτό το ESC, δεν αποτελεί πρόβλημα, καθώς όπως αναφέρθηκε προηγουμένως, το όχημα θα λειτουργήσει σε χαμηλότερη τάση τροφοδοσίας μπαταρίας στα 4S δηλαδή 14.8 Volt. Επίσης ο μικροελεγκτής του κυκλώματος οδήγησης υποστηρίζει το ανοιχτού κώδικα λογισμικό BLHeli, το οποίο προσφέρει πολλά πλεονεκτήματα σε σχέση με άλλες έτοιμες λύσεις της αγοράς. Τα πλεονεκτήματα των ESC που ενσωματώνουν το λογισμικό BLHeli είναι:

- Περισσότερες επιλογές παραμετροποίησης του τρόπου λειτουργίας της γέφυρας (Ανάλογα με τις δυνατότητες του υλικού hardware).
- Ταχύτερη απόκριση, χάρη στη καλύτερη εκμετάλλευση των hardware Interrupt δυνατοτήτων του μικροελεγκτή.
- Εύκολη παραμετροποίηση του λογισμικού μέσω γραφικού περιβάλλοντος από υπολογιστή.
- Ψηφιακά πρωτόκολλα αμφίδρομης επικοινωνίας με δυνατότητα τηλεμετρίας για τα ESC που ο μικροελεγκτής τους τα υποστηρίζει.

Το μοντέλο του ESC που επιλέχθηκε, είναι το Platinum 40A PRO - Version 1 της εταιρίας HobbyWing και τα τεχνικά του χαρακτηριστικά παρουσιάζονται στον Πίνακα 3-4.

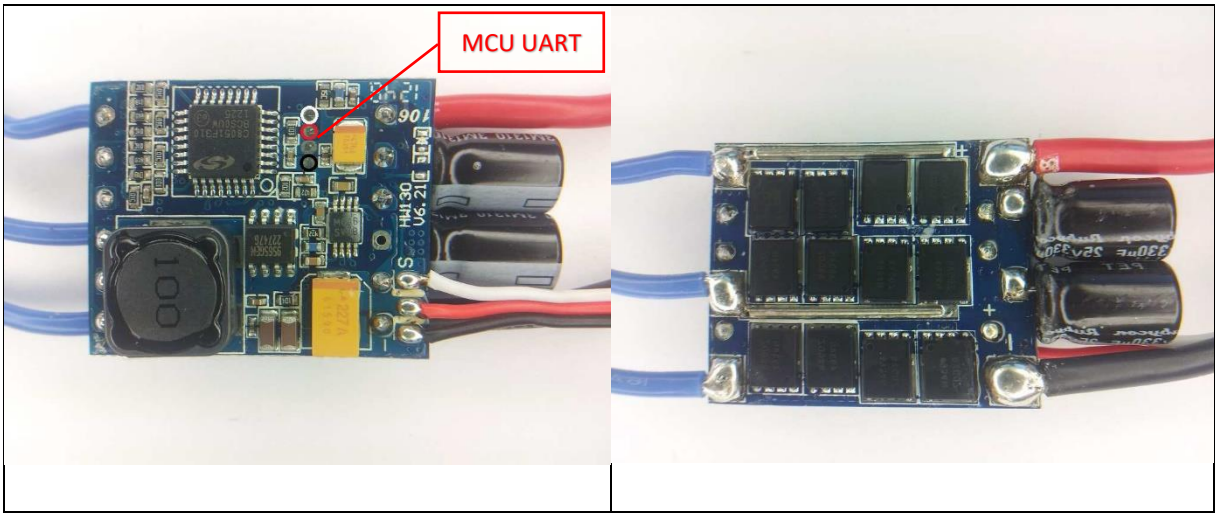
Πίνακας 3-4. Τεχνικά χαρακτηριστικά του κυκλώματος ελέγχου ταχύτητας.[8] [9]

MCU	Silabs C8051FF310
Output	Continuous 40A, burst 60A up to 10 seconds.
Input Voltage	2-6 cells lithium battery or 5-18 cells NiMH battery.
BEC	3A, 5.25V or 6V switchable by software.
Refresh rate of the throttle signal	50Hz to 432Hz.
Control Signal Transmission	PWM, Optical/electrical coupled system.
Max Speed	210000rpm for 2 Poles BLM, 70000rpm for 6 poles BLM, 35000rpm for 12 poles BLM. (BLM = BrushLess Motor)
Size	59mm (L) * 27mm (W) * 12mm (H).
Weight	38g.

Το κύκλωμα οδήγησης που επιλέχθηκε στη παρούσα εργοστασιακή του παραμετροποίηση, δεν ήταν γνωστό εάν υποστήριζε ή είχε ενεργοποιημένες λειτουργίες όπως, έλεγχο ροπής, προστασίες κυκλώματος και ανάστροφη κατεύθυνση στρέψης του ηλεκτροκινητήρα. Ειδικότερα η ανάστροφη κίνηση του μοτέρ είναι μια σημαντική λειτουργία, καθώς είναι απαραίτητη για μανούβρες οπισθοπορίας του σκάφους. Οι λόγοι που αναφέρθηκαν, οδήγησαν στην εγκατάσταση του BLHeli. Στην Εικόνα 3-24 φαίνεται η πλακέτα του κυκλώματος οδήγησης του κινητήρα και πιο συγκριμένα τα pin της σειριακής πόρτας του μικροεπεξεργαστή, για παραμετροποίηση του υλικολογισμικού του.

[8] Specifications: https://www.robotbirds.co.uk/speed-controllers/13-40amp/hobbywing-c-platinum-40a-v1-esc.html

[9] User Manual: https://sekidorc.com/pdf/PlatinumPro_UserManual_en_130502.pdf



Εικόνα 3-24. Η πλακέτα του ESC.

Ακολουθώντας τις οδηγίες της αντίστοιχης ιστοσελίδας Github[10] σχετικά με το BLHeli, εγκαταστάθηκε στο ESC το υλικολογισμικό ανοιχτού κώδικα. Σύμφωνα με τις οδηγίες του Github εκτελέστηκαν τα παρακάτω βήματα:

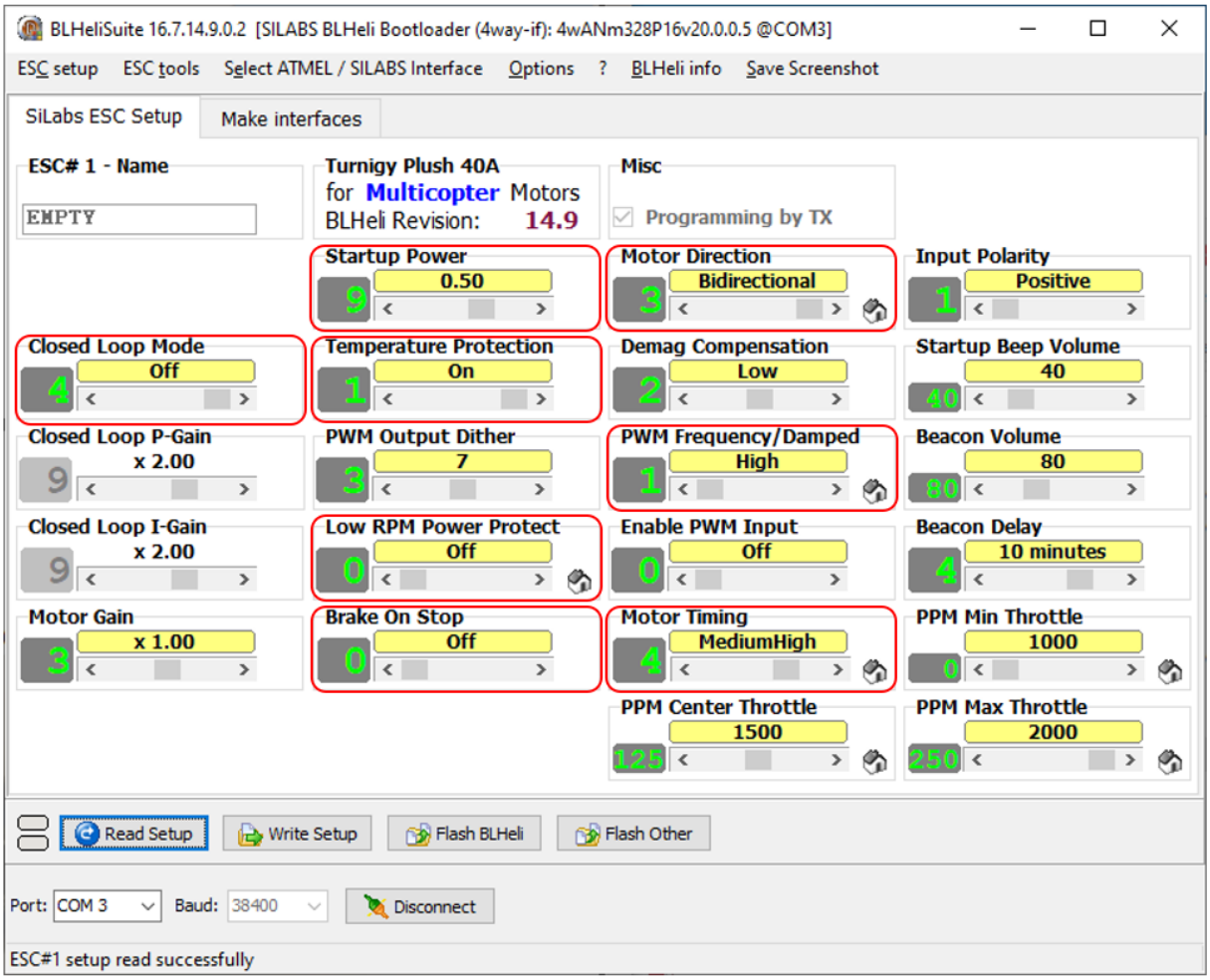
1. Αρχικά αναγνωρίστηκε ο τύπος του ESC με τις οδηγίες του αντίστοιχου εγγράφου[11].
2. Στη συνέχεια κολλήθηκαν προσωρινά καλώδια στα pins (MCU UART) του ESC που διασυνδέονται με τη πλακέτα προγραμματισμού.
3. Τέλος έγινε προετοιμασία της πλακέτας προγραμματισμού και φορτώθηκαν τα υλικολογισμικά (firmware) σύμφωνα με τις οδηγίες του αντίστοιχου εγγράφου[12].

Ως USB to Serial Converter/Programmer χρησιμοποιήθηκε η αναπτυξιακή πλατφόρμα Arduino Nano. Πλέον μέσω υπολογιστή, υπάρχει η δυνατότητα παραμετροποίησης του ESC όπως φαίνεται στις ρυθμίσεις που εφαρμόστηκαν σύμφωνα με την Εικόνα 3-25.

[10] BLHeli Github: https://github.com/bitdump/BLHeli

[11] ESC List: https://github.com/bitdump/BLHeli/blob/master/SiLabs/BLHeli%20supported%20SiLabs%20ESCs.pdf

[12] Firmware Guide: https://github.com/bitdump/BLHeli/blob/master/SiLabs/BLHeli%20programming%20adapters.pdf



Εικόνα 3-25. Το γραφικό περιβάλλον παραμετροποίησης του ESC.

Τέλος, στην Εικόνα 3-25, έχουν τονιστεί οι ρυθμίσεις του ESC που ήταν σημαντικές για αυτή την υλοποίηση. Παρακάτω αναφέρονται περιγραφικά οι λειτουργίες τους και οι λόγοι που τους ανατέθηκαν οι αντίστοιχοι παράμετροι.

- (Closed Loop Mode: off) Η λειτουργία αυτή, όταν είναι ενεργή δίνει προτεραιότητα στην διατήρηση της ροπής και της ταχύτητας περιστροφής του μοτέρ ανεξαρτήτως του φορτίου. Στην εφαρμογή αυτή η λειτουργία δεν χρειάζεται. Αντίθετα εάν ήταν ενεργοποιημένη θα μπορούσε να προκαλέσει ζημιά στη τριφασική γέφυρα, καθώς το μέγιστο ρεύμα τροφοδοσίας του κινητήρα μπορεί να ξεπεράσει το μέγιστο ρεύμα φορτίου του ESC.
- (Startup Power: 0.50) Η παράμετρος αυτή, καθορίζει το συντελεστή περιορισμού του ρεύματος εκκίνησης για την αποφυγή υπερφορτώσεων του κινητήρα και του κυκλώματος οδήγησης. Στην υλοποίηση αυτή το ρεύμα περιορίστηκε στο μισό με συντελεστή 0.5.
- (Temperature Protection: On) Ενεργοποίηση του κυκλώματος θερμικής προστασίας για το ESC.
- (Low RPM Power Protect: Off) Η παράμετρος αυτή περιορίζει το ρεύμα εκκίνησης σε μικρούς κινητήρες με πολύ μεγάλες ταχύτητες περιστροφής. Αυτή η προστασία δεν χρειάζεται σε μεγάλους αργόστροφους κινητήρες όπως σε αυτή την υλοποίηση, διότι δημιουργούνται προβλήματα ισχύος και συγχρονισμού.

- (Brake On Stop: Off) Η λειτουργία αυτή όταν είναι ενεργοποιημένη φρενάρει τον κινητήρα μόλις το εισερχόμενο σήμα ελέγχου του ESC αντιστοιχεί σε μηδενική ταχύτητα. Στην περίπτωση αυτή, δε χρειάζεται να φρενάρει η προπέλα του σκάφους για εξοικονόμηση ενέργειας και ελαχιστοποίηση των τριβών με το νερό.
- (Motor Direction: Bidirectional) Μέσω αυτής της επιλογής μπορεί να επιλεχθεί η φορά περιστροφής του κινητήρα, μονής κατεύθυνσης αριστερόστροφη ή δεξιόστροφη ή και διπλής κατεύθυνσης. Καθώς το σκάφος χρειάζεται να κάνει μανούβρες οπισθοπορίας επιλέχθηκε η παράμετρος «Bidirectional».
- (PWM Frequency/Damped: High) Η ρύθμιση αυτή, καθορίζει τη συχνότητα του PWM σήματος για τον έλεγχο των Mosfet της τριφασικής γέφυρας του ESC. Για τη παρούσα υλοποίηση, οι πειραματισμοί έδωσαν καλύτερη απόκριση ελέγχου και ισχύος κινητήρα στην επιλογή «High».
- (Motor Timing: MediumHigh) Στη παράμετρο αυτή επιλέγονται προκαθορισμένες σταθερές χρόνου για την εναλλαγή των φάσεων του κινητήρα. Η επιλογή «MediumHigh» μέσω πειραματισμών, έδωσε καλύτερα αποτελέσματα συγχρονισμού του κινητήρα, αυξάνοντας τη ροπή.

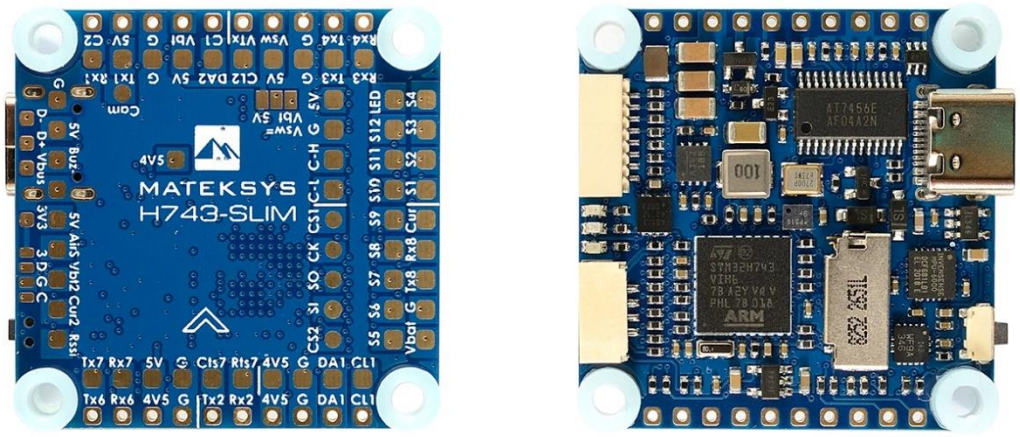
3.4.3 Αυτόματος πιλότος και περιφερειακά υποσυστήματα

Το μη επανδρωμένο σκάφος επιφανείας, χρησιμοποιεί ένα ενσωματωμένο σύστημα που συνδυάζει μικροελεγκτές και αισθητήρες, για τον έλεγχο της πλοήγησης του, γνωστό και ως Flight Controller ή «ελεγκτή πτήσης». Ο ελεγκτής πτήσης που επιλέχθηκε, στο υλικό του μέρος, έχει σχεδιαστεί ώστε να είναι συμβατός με μία πληθώρα υλικολογισμικών (Firmware) ανοικτού κώδικα, συμπεριλαμβανομένου και του Ardupilot. Η ονομασία «Flight Controller», προέρχεται από το γεγονός ότι τα περισσότερα από αυτά τα υλικολογισμικά αναπτύχθηκαν αρχικά ή εξειδικεύονται, στον έλεγχο ιπτάμενων μη επανδρωμένων οχημάτων. Το Ardupilot που θα χρησιμοποιηθεί στη συγκεκριμένη υλοποίηση, όπως αναφέρθηκε στην ενότητα 2.6, είναι το πιο ανεπτυγμένο στη κατηγορία του, διαθέτει αυτόματη πλοήγηση και μπορεί να χρησιμοποιηθεί σε διάφορους τύπους μη επανδρωμένων οχημάτων, συμπεριλαμβανομένων και των σκαφών επιφανείας. Επιπλέον, μέσω του ανοιχτού πρωτόκολλου επικοινωνίας (MAVLink) που υποστηρίζει, επιτρέπει στο υλικό μέρος να διασυνδεθεί με άλλα συμβατά συστήματα για την επέκταση των δυνατοτήτων του. Στη συγκεκριμένη υλοποίηση, χρησιμοποιήθηκε η πλακέτα αυτόματου πιλότου H743-SLIM από την Matek Systems. Η συγκεκριμένη πλακέτα, χάρη στον μικροελεγκτή που ενσωματώνει διαθέτει ένα μεγάλο αριθμό εισόδων, εξόδων και διαύλων επικοινωνίας για τη διασύνδεση διαφόρων τύπων υποσυστημάτων ανάλογα την εκάστοτε εφαρμογή.

Το ενσωματωμένο σύστημα H743-SLIM, που επιλέχθηκε στη παρούσα υλοποίηση, διαθέτει αρκετούς ενσωματωμένους αισθητήρες για την πλοήγηση του σκάφους. Ειδικότερα, ο αριθμός των αισθητήρων είναι δύο IMU (Inertia Measurement Unit) και ένα βαρόμετρο.

Ο συνδυασμός των δεδομένων, από ένα γυροσκόπιο και ένα επιταχυνσιόμετρο, μπορεί με τη κατάλληλη επεξεργασία αλγορίθμων από το μικροελεγκτή, να δώσει τη στάση ενός οχήματος στο τρισδιάστατο χώρο. Η ενσωμάτωση, αυτών των δύο αισθητήρων σε ένα ολοκληρωμένο κύκλωμα, αποτελεί μια μονάδα διάταξης γνωστής και ως IMU, που αποτελεί βασικό εξάρτημα ενός συστήματος πλοήγησης. Για να εξασφαλιστεί η εφεδρεία και η αξιοπιστία των δεδομένων που παράγονται από αυτούς τους αισθητήρες, συνδυάζονται οι μετρήσεις από πολλαπλές μονάδες IMU, όπως μπορεί να συμβεί στη συγκεκριμένη ηλεκτρονική διάταξη.

Αντίστοιχα, το βαρόμετρο που ενσωματώνεται, μπορεί να δώσει πληροφορίες για το υψόμετρο του οχήματος και τη θερμοκρασία περιβάλλοντος [47] [48] [49].



Εικόνα 3-26. Η πλακέτα του Flight Controller. Μπροστινή και πίσω όψη.

Στον Πίνακα 3-5, παρουσιάζονται τα τεχνικά χαρακτηριστικά του συστήματος πλοήγησης της παρούσας υλοποίησης.

Πίνακας 3-5. Τεχνικά χαρακτηριστικά του Flight Controller.[13]

Board Specifications
Manufacturer: Matek Systems
Model: H743-SLIM
MCU: STM32H743VIH6, 480MHz , 1MB RAM, 2MB Flash
Board V1.0 IMU: MPU6000 (SPI1) & ICM20602 (SPI4)
Board V1.5 IMU: MPU6000 (SPI1) & ICM42605 (SPI4)
Barometer: Infineon DPS310 (I2C2)
OSD: AT7456E (SPI2)
Blackbox: MicroSD card socket (SDIO)
7x UART: (1,2,3,4,6,7,8) with built-in inversion.
13x PWM outputs(including “LED” pad)
2x I2C
1x CAN
6x ADC (VBAT, Current, RSSI, Analog AirSpeed, Vbat2, Cur2)
3x LEDs for FC STATUS (Blue, Red) and 3.3V indicator(Red)
1x SPI3 breakout
USB Type-C(USB2.0)

[13] Flight Controller Datasheet: http://www.mateksys.com/?portfolio=h743-slim#tab-id-2

1x JST-SH1.0_8pin connector (Vbat/G/Curr/Rx8/S1/S2/S3/S4)	
1x JST-GH1.25_4pin connector (5V/CAN-H/CAN-L/G)	
Dual Camera Inputs switch	
5V/Vbat filtered power ON/OFF switch	
DJI FPV OSD is supported by any spare UART	
Power Specifications	
Vbat Input	6~36V (2~8S LiPo)
BEC	5V 2A cont. (Max.3A)
LDO 3.3V	Max.200mA
Static power	200mA@5V with Betaflight, 150mA@5V with ArduPilot
No Current Sensor built-in	
ADC Vbat2 pad supports Max. 69V (voltage divider: 1K:20K)	
Flight Controller Board Compatible Firmware	
ArduPilot(ChiBiOS)	MATEKH743
BetaFlight	MATEKH743
INAV	MATEKH743
Physical Dimensions	
Mounting	30.5 x 30.5mm, Φ4mm with Grommets Φ3mm
Dimensions	36 x 36 x 5 mm
Weight	7g

Απαραίτητη ήταν, η εγκατάσταση του λογισμικού Ardupilot στον ελεγκτή H743-SLIM, που έγινε σύμφωνα με τα ακόλουθα βήματα:

1. Εγκατάσταση του προγράμματος STM32CubeProgrammer[14].
2. Λήψη του υλικολογισμικού[15] με όνομα “ardurover_with_bl.hex” από την ιστοσελίδα του Ardupilot για τον ελεγκτή H743-SLIM. Η συγκεκριμένη έκδοση ειδικεύεται στον έλεγχο επίγειων οχημάτων “Rover”, όπου ανήκει το σκάφος επιφανείας της υλοποίησης.
3. Διαγραφή του δοκιμαστικού εργοστασιακού υλικολογισμικού από τη μνήμη “flash” του μικροελεγκτή και εγκατάσταση του νέου, σύμφωνα με τις οδηγίες[16] του κατασκευαστή Matek Systems για τον ελεγκτή H743-SLIM.

[14] STM32CubeProgrammer link: https://www.st.com/en/development-tools/stm32cubeprog.html

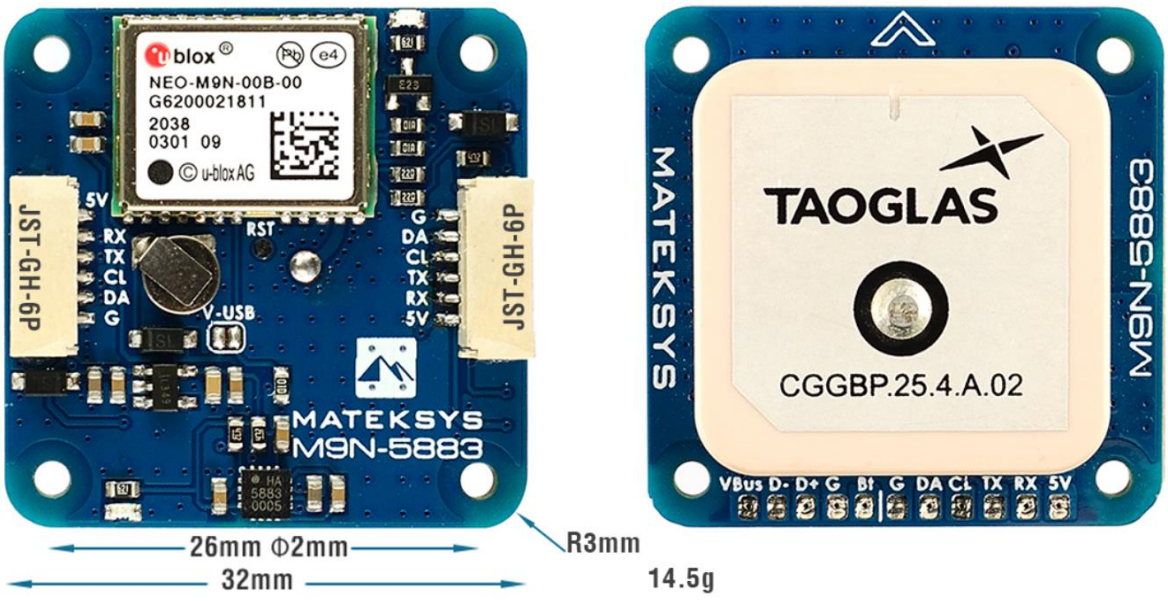
[15] Ardupilot Boat firmware link: https://firmware.ardupilot.org/Rover/stable-4.1.0/MatekH743/

[16] H743-SLIM flashing guide: http://www.mateksys.com/?p=6905

Επίσης, γενικές οδηγίες εγκατάστασης του firmware σε αντίστοιχα συμβατά ενσωματωμένα συστήματα δίνονται από τον επίσημο οδηγό[17] της ιστοσελίδας του Ardupilot.

3.4.4 Δέκτης γεωδαιτικών δεδομένων GNSS

Για να γνωρίζει το μη επανδρωμένο όχημα τη θέση του στο γεωγραφικό χώρο, με τη μορφή συντεταγμένων, απαραίτητη ήταν η προσθήκη ενός δέκτη του παγκοσμίου δορυφορικού συστήματος πλοήγησης GNSS. Επιπρόσθετα, για τον προσανατολισμό του οχήματος χρειάστηκε να προστεθεί ένα μαγνητόμετρο που επιτελεί το ρόλο της πυξίδας. Το μαγνητόμετρο QMC5883L και ο GNSS δέκτης NEO-M9N που επιλέχθηκαν στη παρούσα υλοποίηση, ενσωματώνονται σε μία πλακέτα με κοινή τροφοδοσία και συνδέονται με τον αυτόματο πιλότο μέσω των αντίστοιχων διαύλων επικοινωνίας. Πιο συγκεκριμένα οι δίαυλοι αυτοί αντιστοιχούν σε μία σειριακή θύρα UART για τον δέκτη GNSS και τον δίαυλο I2C για το μαγνητόμετρο. Η προσθήκη του QMC5883L σε συνδυασμό τον ελεγκτή H743-SLIM και τα IMU που ενσωματώνει, συνθέτουν ένα ολοκληρωμένο σύστημα πλοήγησης AHRS, καθώς τα δεδομένα από αυτούς τους αισθητήρες επεξεργάζεται ο μικροελεγκτής για την εξαγωγή της στάσης και του προσανατολισμού του οχήματος στο τρισδιάστατο χώρο. Τα δεδομένα του AHRS (Attitude and Heading Reference System) που προκύπτουν από τον μικροελεγκτή μέσω αλγορίθμων ενοποίησης αισθητήρων (Sensor Fusion), ενισχύσουν την αξιοπιστία και την ακρίβεια του στίγματος από τον δέκτη GNSS για τον υπολογισμό της θέσης του σκάφους [50] [51] [52]. Η προσθήκη του δέκτη GNSS και της πυξίδας, είναι απαραίτητη για την εκμετάλλευση των δυνατοτήτων αυτόνομης πλοήγησης που παρέχει ο Ardupilot.



Εικόνα 3-27. Η πλακέτα του δέκτη GNSS με το ενσωματωμένο μαγνητόμετρο. Μπροστινή και πίσω όψη.

[17] Ardupilot guide: https://ardupilot.org/copter/docs/common-loading-firmware-onto-chibios-only-boards.html

Η ηλεκτρονική διάταξη GNSS που επιλέχθηκε για το όχημα είναι συμβατό εξάρτημα επέκτασης του ελεγκτή πλοήγησης H743-SLIM και τα τεχνικά χαρακτηριστικά του παρουσιάζονται στον Πίνακα 3-6.

Πίνακας 3-6. Τεχνικά χαρακτηριστικά του δέκτη γεωδαιτικών δεδομένων.[18] [19] [20]

Specifications	
Manufacturer	Matek Systems
Model	M9N-5883
GNSS u-blox NEO-M9N (GPS, GLONASS, Galileo and BeiDou)	
Position accuracy	2.0m CEP
Nav. Update rate	25Hz
UART baudrate	38400 default
Operating Temperatures	-20~80 °C
Receiver sensitivity	
Tracking and navigation	-167 dBm
Reacquisition	-160 dBm
Cold start	-148 dBm
Hot start	-159 dBm
Magnetic Compass QMC5883L	
Patch antenna size	25*25*4mm
Power	
Input voltage range	4~5.5V (5V pad/pin)
Power consumption	50mA
Connectivity	
UART(TX, RX) interface for GNSS NEO-M9N	
I2C(DA, CL) interface for Compass QMC5883L	
JST-GH-6P connector	
Indicators	
3.3V Power LED, Red	
GNSS PPS LED, Green, blinking(1Hz) when GNSS has 3D fixed	
Physical dimensions	
Dimensions	32mm*32mm*10mm
Module Weight	14.5g

[18] Module product page: http://www.mateksys.com/?portfolio=m9n-5883#tab-id-2

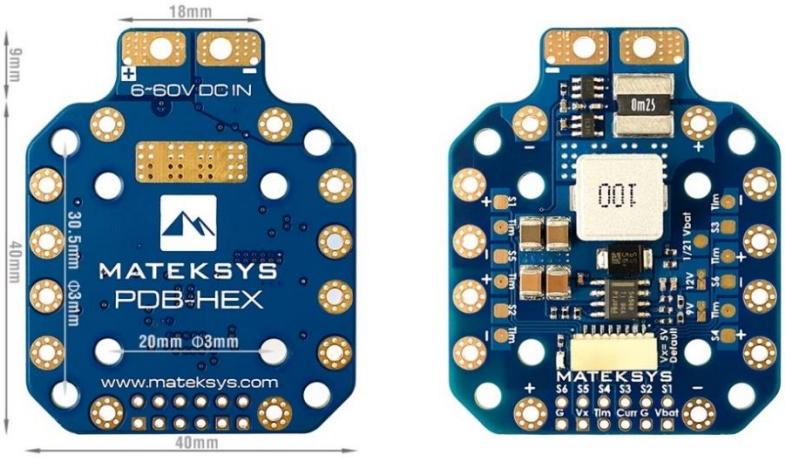
[19] NEO-M9N GNSS Datasheet: https://www.u-blox.com/en/product/neo-m9n-module#tab-documentation-resources

[20] QMC5883L Datasheet: https://www.filipeflop.com/img/files/download/Datasheet-QMC5883L-1.0%20.pdf

3.4.5 Πλακέτα διανομής ισχύος

Ο διανομέας ισχύος PDB (Power Distribution Board) είναι μια απαραίτητη ηλεκτρονική διάταξη υπεύθυνη για τη διανομή ισχύος από μία ή περισσότερες πηγές ενέργειας στα επιμέρους ηλεκτρονικά συστήματα του μη επανδρωμένου οχήματος. Επιπλέον ο διανομέας είναι εξοπλισμένος με αισθητήρες για την αποστολή μετρήσεων σχετικά με τη κατάσταση των πηγών και των ενεργειακών τους αποθεμάτων στο σύστημα πλοήγησης. Επίσης μια διάταξη PDB ενσωματώνει μετατροπείς τάσεως για την μεταφορά ενέργειας μεταξύ διάφορων υποσυστημάτων που μπορεί να εγκατασταθούν αλλά και αναμονές σύνδεσης για την τροφοδοσία των κινητήρων και των ενεργοποιητών του οχήματος.

Για το σκάφος της διατριβής επιλέχθηκε μια συμβατή διάταξη PDB με τον ελεγκτή πλοήγησης H743-SLIM από τον ίδιο κατασκευαστή. Η πλακέτα είναι σχετικά απλή στη κατασκευή της, ενσωματώνοντας δύο αναλογικούς αισθητήρες για τη μέτρηση του ρεύματος και της τάσης μίας πηγής ενέργειας. Η διάταξη διαθέτει ένα μετατροπέα συνεχούς ρεύματος (DC σε DC) για την τροφοδοσία άλλων περιφερειακών συσκευών χαμηλής ισχύος. Επιπρόσθετα ενσωματώνονται αναμονές (Pads) για τη τροφοδοσία των κυκλωμάτων ισχύος κινητήρων και ενεργοποιητών καθώς και αναμονών για τη μεταφορά τροφοδοσίας και σημάτων από και προς τον ελεγκτή H743-SLIM. Στον Πίνακα 3-7 παρουσιάζονται τα τεχνικά χαρακτηριστικά του συγκεκριμένου PDB.



Εικόνα 3-28. Ο διανομέας ισχύος από τις δύο όψεις του PCB.

Πίνακας 3-7. Τεχνικά χαρακτηριστικά διανομέα ισχύος.[21]

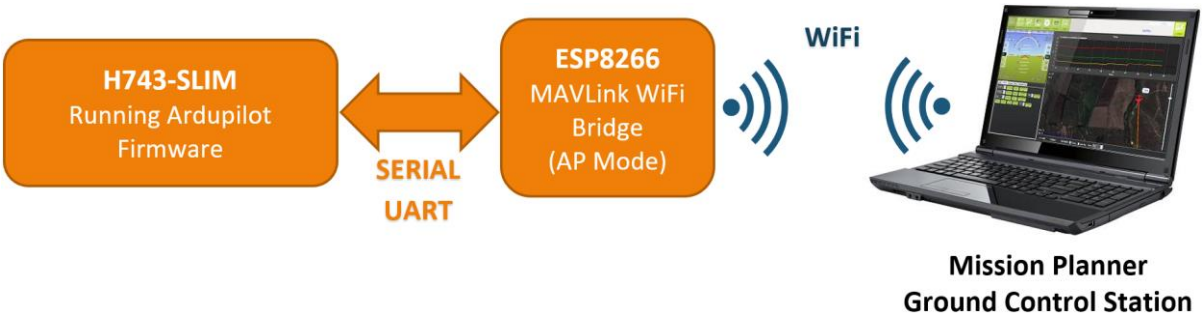
PDB Specifications	
Manufacturer	Matek Systems
Model	PDB-HEX, 12S
Input voltage range	6~60V (2~12S LiPo)
PDB/current sense resistor	140A cont, 264A burst.
ESC power pads	60A continuous. 100A burst

[21] PDB board specifications: http://www.mateksys.com/?portfolio=pdb-hex#tab-id-2

6x ESC power /signal/ telemetry pads	
Current Senor	264A, 3.3V ADC
Current Scale 125(INAV/BF), BATT_AMP_PERVLT 80 (ArduPilot)	
JST-SH-8P Connector and Pads	
Vbat	Battery voltage
G	Ground
Curr	Current signal
Tlm	Duplicates Tlm pads for ESC telemetry
S1/S2/S3/S4/S5/S6	ESC control signal
Vx	Regulator output
BEC/DC to DC converter Vx output (5V default)	
Vx= 5V by default, Continuous 5 Amps	
Bridge 9V jumper, Vx= 9V, Continuous 4 Amps, Max.5A	
Bridge 12V jumper, Vx= 12V, Continuous 4 Amps, Max.5A	
Output Short-circuit tolerant (1 seconds)	
Overcurrent protection & self-recovery	
Voltage divider for 9-12S battery voltage sensing	
1K:20K resistors	
Output 1/21 Vbat voltage	
Physical	
Mounting	30.5 x 30.5mm, 20 x 20mm, Φ3mm
Dimensions	49 x 40 x 6 mm
Weight	12g

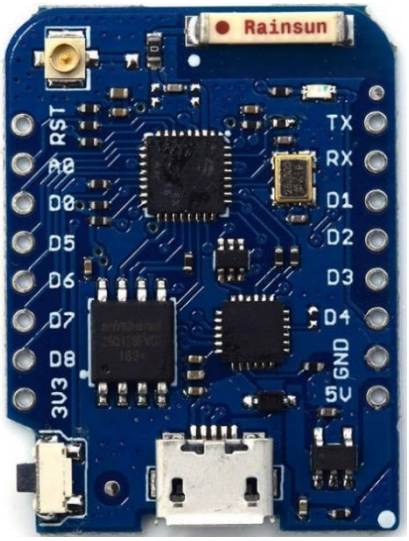
3.4.6 Περιφερειακό σύστημα τηλεμετρίας μέσω μικροελεγκτή ESP8266

Για να μπορεί να γίνει εποπτεία των λειτουργιών του αυτόματου πιλότου, κατά τη διάρκεια των δοκιμών και για να υπάρχει η δυνατότητα αντιμετώπισης πιθανών προβλημάτων, που μπορεί να μην επιλύονται με το σύστημα τηλεχειρισμού, ήταν απαραίτητη η προσθήκη ενός υποσυστήματος τηλεμετρίας. Πιο συγκεκριμένα το υποσύστημα αυτό, αποτελείται από τον μικροελεγκτή ESP8266, ο οποίος με το κατάλληλο πρόγραμμα ονομαζόμενο “MAVLink WiFi Bridge”, λειτουργεί σαν ένας διακομιστής που δίνει πρόσβαση σε μία σειριακή πόρτα του αυτόματου πιλότου μέσω του πρωτόκολλου WiFi. Ως εκ τούτου, με τη χρήση ενός υπολογιστή που διαθέτει ασύρματη κάρτα δικτύου WiFi και του κατάλληλου προγράμματος διαχείρισης (Mission Planner) μπορεί να γίνει άμεση διαχείριση του Ardupilot που ελέγχει τη πλοήγηση του σκάφους απομακρυσμένα, όπως παρουσιάζεται στην Εικόνα 3-29.



Εικόνα 3-29. Το σύστημα τηλεμετρίας μέσω ESP8266.

Για τη παρούσα εφαρμογή χρησιμοποιήθηκε η αναπτυξιακή πλατφόρμα D1 mini Pro Version 1 της WEMOS electronics καθώς έχει χαμηλό κόστος και δυνατότητα προσάρτησης εξωτερικής κεραίας για την ενίσχυση της εμβέλειας του WiFi σήματος, όπως φαίνεται στην Εικόνα 3-30. Στον Πίνακα 3-8 παρουσιάζονται τα τεχνικά χαρακτηριστικά τις αναπτυξιακής πλατφόρμας.



Εικόνα 3-30. Το αναπτυξιακό WEMOS D1 mini Pro V1.

Πίνακας 3-8. Τεχνικά χαρακτηριστικά αναπτυξιακού D1 mini Pro (V1).[22]

Specifications	
Manufacturer	WEMOS
Model	D1 mini Pro (Version 1)
Microcontroller	ESP-8266EX[23]
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1(Max input 3.2V) pin A0 (ADC0)
Interrupts	All pins except pin D0 (GPIO16)

[22]Board Specifications: https://grobotronics.com/wemos-d1-mini-pro-esp8266-v1.0.html?sl=en

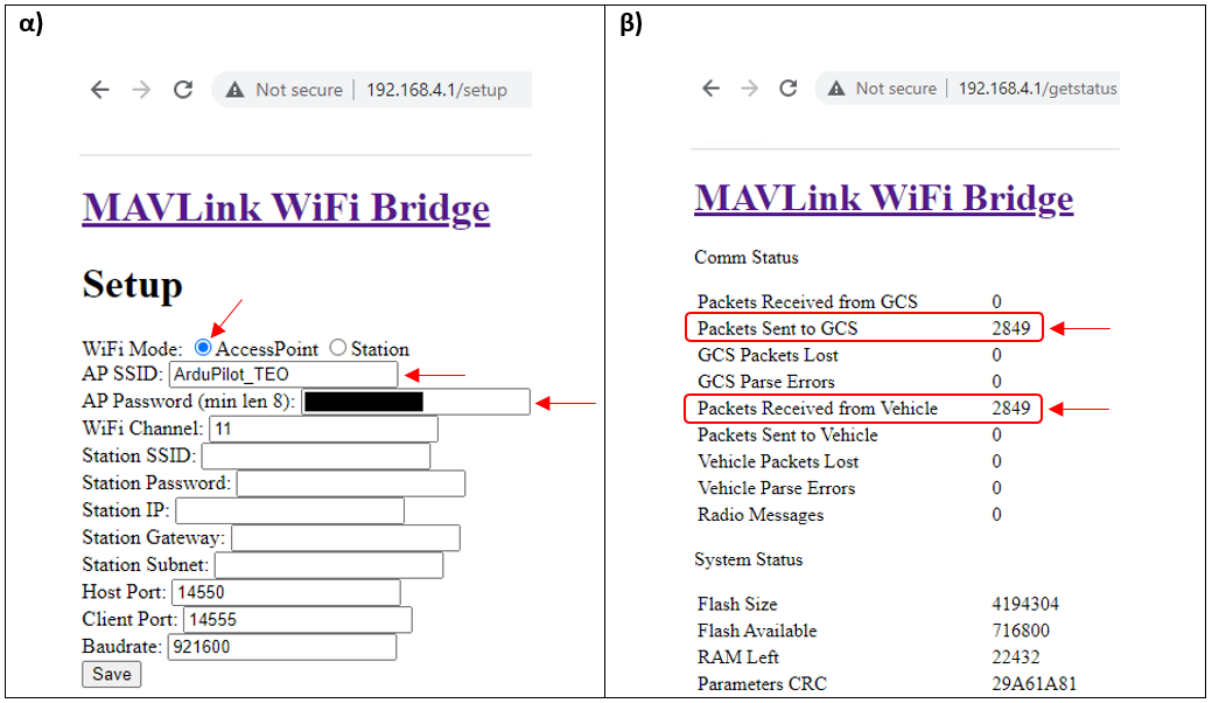
[23]MCU datasheet: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

PWM/I2C/one-wire (implemented in software)	
SPI (Pins D5:SCK, D6:MISO, D7:MOSI, D8:SS)	
UART (Pins TX,RX)	
Clock Speed	80MHz/160MHz
Flash	16M bytes
External antenna connector	
Built-in ceramic antenna	
CP2104 USB-TO-UART IC	
Dimensions	34.2x25.6x4mm
Weight	2.5g

Για τη λειτουργία της τηλεμετρίας, ακολουθήθηκαν τα παρακάτω βήματα με την αντίστοιχη σειρά:

1. Εγκαταστάθηκε το υλικολογισμικό σύμφωνα με τις οδηγίες[24] της σελίδας του Ardupilot για την προετοιμασία του ESP8266 και έγιναν οι απαραίτητες συνδέσεις με τη UART του ελεγκτή πλοήγησης (υποενότητα 3.4.9).
2. Στη συνέχεια τροφοδοτήθηκε το αναπτυξιακό (D1 mini Pro), το οποίο στις προεπιλεγμένες ρυθμίσεις του λειτουργεί ως σημείο πρόσβασης (Access Point).
3. Από ένα υπολογιστή με WiFi δυνατότητες, έγινε πρόσβαση στο δίκτυο της συσκευής με SSID “Ardupilot” και Password “ardupilot”.
4. Μέσα από ένα περιηγητή διαδικτύου αποκτήθηκε πρόσβαση στην IP διεύθυνση της συσκευής 192.168.1.4 όπου εμφανίζεται το αντίστοιχο γραφικό περιβάλλον διαχείρισης. Στο σύνδεσμο “Setup” τροποποιήθηκαν τα SSID και Password για λόγους ασφαλείας, όπως στην Εικόνα 3-31α.
5. Τέλος επιβεβαιώνεται η επικοινωνία του υπολογιστή με το σύστημα τηλεμετρίας και τον Ardupilot του σκάφους, μέσω του συνδέσμου “Status”. Στη σελίδα αυτή θα πρέπει να παρατηρείται αλλαγή στη κυκλοφορία των πακέτων δεδομένων (με ανανέωση σελίδας) από και προς το όχημα, όπως στην Εικόνα 3-31β.

[24]Ardupilot ESP8266 telemetry guide: https://ardupilot.org/rover/docs/common-esp8266-telemetry.html



Εικόνα 3-31 Το γραφικό περιβάλλον της γέφυρας τηλεμετρίας.

3.4.7 Περιφερειακό σύστημα ελέγχου μέσω μικροελεγκτή ESP32

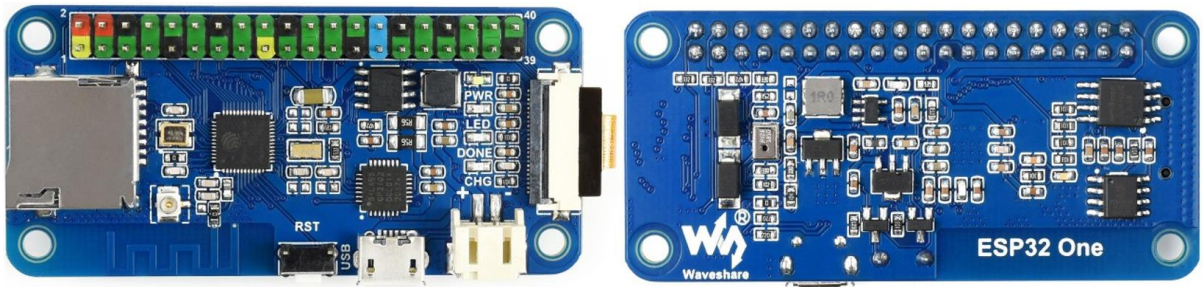
Το σημαντικότερο υποσύστημα του μη επανδρωμένου σκάφους σε αυτή την υλοποίηση αποτελεί ο μικροελεγκτής ESP32 που συνδέεται με τον αυτόματο πιλότο μέσω σειριακής θύρας. Ο ρόλος αυτού του μικροελεγκτή είναι, η γεφύρωση του πρωτόκολλου MAVLink που χρησιμοποιεί ο αυτόματος πιλότος με το διαδικτυακό πρωτόκολλο MQTT.

Εικόνα 3-32. Το σύστημα ελέγχου του Ardupilot μέσω του ESP32.

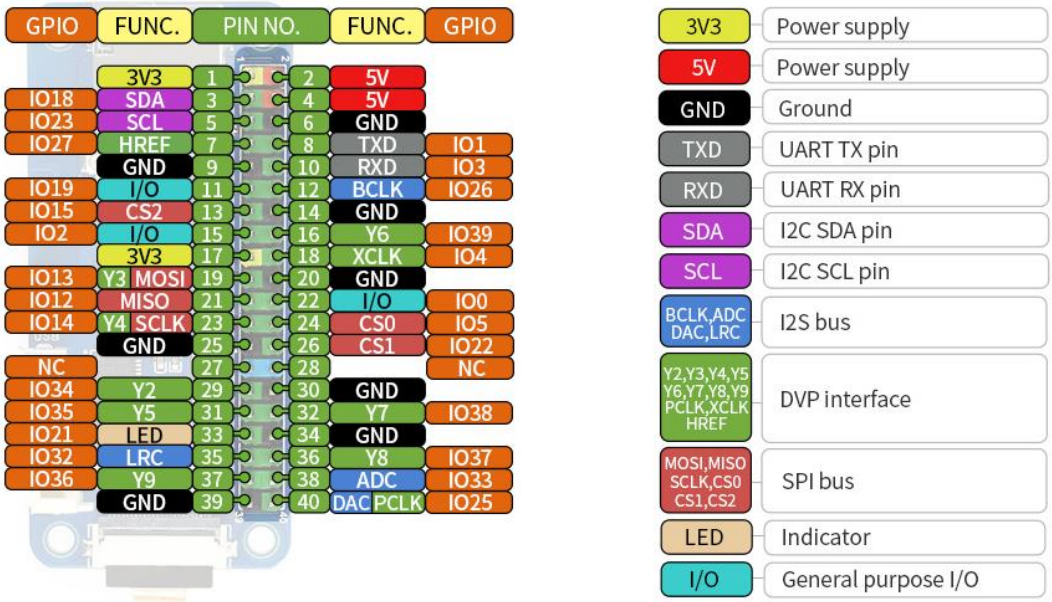
Η διασύνδεση του μικροελεγκτή με το διαδίκτυο επιτυγχάνεται μέσω των WiFi δυνατοτήτων που ενσωματώνει. Πιο συγκεκριμένα, το αναπτυξιακό ESP32 έχει προγραμματιστεί έτσι ώστε να παρουσιάζεται στον ελεγκτή του Ardupilot, ως περιφερειακός υπολογιστής πλοήγησης, διατηρώντας ένα “Heartbeat” με περίοδο ενός δευτερολέπτου, ως ένδειξη ότι είναι ενεργός. Το “Heartbeat” είναι σημαντικό γιατί κρατάει ενεργή τη σύνδεση MAVLink μέσω του σειριακού διαύλου UART που ενώνει τις δύο συσκευές. Επίσης από την έναρξη λειτουργίας του, το ESP32 ζητάει από το σύστημα πλοήγησης μια ροή δεδομένων σχετικά με όλες τις παραμέτρους που αφορούν τη κατάσταση του οχήματος και την αποστολή του, με περίοδο ένα δευτερόλεπτο. Από αυτό τον όγκο δεδομένων, επιλέγονται οι πιο σημαντικές πληροφορίες που εξυπηρετούν τις ανάγκες της παρούσας υλοποίησης και στη συνέχεια χωρίζονται σε MQTT μηνύματα που αποστέλλονται (publish) μέσω WiFi στα αντίστοιχα topics του MQTT Broker.

Για τον έλεγχο του οχήματος μέσω της διαδικτυακής πλατφόρμας υπηρεσιών, το ESP32 κάνει subscribe στα topics που σχετίζονται με εντολές που επιλέχθηκαν να εκτελέσει το παρών όχημα, στα πλαίσια της εργασίας. Αντίστοιχα, δημιουργήθηκε αλγόριθμος που μεταφράζει τα MQTT μηνύματα των topics-εντολών σε εντολές του πρωτοκόλλου MAVLink που αποστέλλονται μέσω UART στο σύστημα πλοήγησης για να τις εκτελέσει. Επιπρόσθετα, παρόμοια με τις μεθόδους που περιεγράφηκαν προηγουμένως, το ESP32 αναλαμβάνει να μεταφράσει τα “Acknowledgements” που επιστρέφει ο Ardupilot, σχετικά με τις εντολές που του ανατέθηκαν, για την ενημέρωση των αντίστοιχων MQTT topics.

Για την παρούσα εφαρμογή, χρησιμοποιήθηκε η πλατφόρμα ανάπτυξης ESP32 ONE της Waveshare, καθώς ήταν ήδη διαθέσιμη από προηγούμενη κατασκευή. Επίσης διαθέτει σύνδεσμο για προσάρτηση εξωτερικής κεραίας σε περίπτωση που είναι επιθυμητή η επέκταση της εμβέλειας του WiFi σήματος. Παρακάτω παρουσιάζονται εικόνες και τα τεχνικά χαρακτηριστικά του αναπτυξιακού.



Εικόνα 3-33. Μικροελεγκτής ESP32-ONE.



Εικόνα 3-34. Pin out διάγραμμα μικροελεγκτή ESP32 ONE.

Πίνακας 3-9. Τεχνικά χαρακτηριστικά ESP32 ONE. [25]

General Specifications	
Manufacturer	Waveshare
Model	ESP32 ONE
Processor	Xtensa LX6 dual-core processor @240MHz
Wifi	802.11b/g/n
Bluetooth	Bluetooth 4.2, including traditional BR/EDR, and BLE low-energy
Camera	OV2640 (UXGA 1622×1200) (optional)
SRAM	520KB+8MB
Flash	448KB+4MB
Power Specifications	
Input Voltage	3.3V~5V
Operating Voltage	3.3V
Charge Current	1A@5V
Overall Consumption	7.7mA@5V (ESP32 under deep sleep)
Physical Specifications	
Dimensions	65mm × 30.5mm
Operating Temperature	-40℃ ~ 85℃
Extra Features	
Digital microphone, supports applications like voice processing	
Standard 40PIN GPIO header, compatible with Raspberry Pi HATs for sorts of IoT applications	
Integrated Li-ion battery boost charging manager, supports battery charging/discharging, can also be powered from USB connection	
Compatible with Arduino and ESP-IDF software SDK, seamlessly connecting with ESP-WHO applications	
IPEX antenna connector	
Onboard SD extension	
CP2102 USB to UART converter, for serial debugging/programming	
4x LED indicators, to check the operating status	

3.4.8 Σύστημα Τηλεκατεύθυνσης

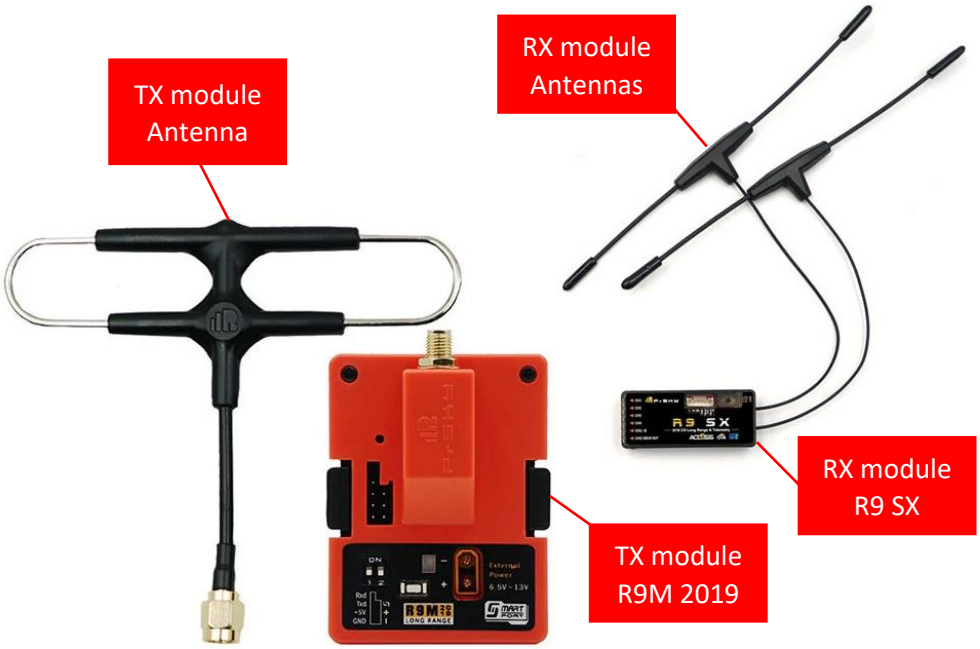
Για τον άμεσο έλεγχο από τον χειριστή, τη προσθήκη ενός εφεδρικού μέσου ελέγχου και την αποφυγή ατυχημάτων στη περίπτωση ενός απρόβλεπτου συμβάντος, ενσωματώθηκε στο μη επανδρωμένο σκάφος ένα σύστημα τηλεχειρισμού. Με τις κατάλληλες ρυθμίσεις στον

[25] ESP32 ONE specifications link: https://www.waveshare.com/esp32-one.htm

ελεγκτή H743-SLIM του Ardupilot, μπορεί μέσω του ανθρώπινου παράγοντα, να γίνει άμεση διορθωτική παρέμβαση στη πορεία του οχήματος ή και ακύρωση της τρέχουσας αποστολής εάν κριθεί απαραίτητο. Το σύστημα αυτό, είναι ένα ολοκληρωμένο προϊόν από την εταιρία FrSky που αποτελείται από:

- ένα δέκτη,
- ένα πομπό,
- και ένα τηλεχειριστήριο.

Τα εξαρτήματα είναι σχεδιασμένα να είναι συμβατά με το πρωτόκολλο ασύρματης μετάδοσης ACCESS. Παρόλο που το τηλεχειριστήριο που επιλέχθηκε, διαθέτει ενσωματωμένο πομποδέκτη, η μπάντα συχνοτήτων εκπομπής του περιορίζεται στα 2.4GHz και για αυτό δεν θα χρησιμοποιηθεί σε αυτή την εφαρμογή. Ο λόγος για αυτή την επιλογή είναι ότι στο σκάφος ενσωματώνονται δύο συσκευές που χρησιμοποιούν το πρωτόκολλο WiFi. Οι συσκευές αυτές είναι το ESP8266 και το ESP32 που αναφέρθηκαν στις προηγούμενες ενότητες 0 και 3.4.7. Η παρουσία πολλών συστημάτων ασύρματης μετάδοσης τοποθετημένες σε κοντινή απόσταση μεταξύ τους, δυσχεραίνει τη ποιότητα των τηλεπικοινωνιών στον χώρο εμβέλειας τους, ειδικότερα όταν αυτά εκπέμπουν στις ίδιες ή γειτονικές συχνότητες [53]. Επιλέχθηκε επομένως ένα σετ πομπού και δέκτη που λειτουργούν στη ζώνη των ISM συχνοτήτων στα 900MHz. Το μοντέλο του πομπού είναι το R9M 2019, ενώ του δέκτη το R9 SX και παρουσιάζονται στην Εικόνα 3-35.



Εικόνα 3-35. Σύστημα πομπού και δέκτη της FrSky για την ISM μπάντα των 900MHz.

Το τηλεχειριστήριο που επιλέχθηκε είναι το μοντέλο Taranis QX7 ACCESS που παρουσιάζεται στην Εικόνα 3-36α. Ο πομπός R9M 2019 ενσωματώνεται στην πίσω πλευρά του χειριστηρίου όπως απεικονίζεται στην Εικόνα 3-36β.

α)

β)

Εικόνα 3-36. Το τηλεχειριστήριο QX7 με τον πομπό R9M 2019.

Τα τεχνικά χαρακτηριστικά των επιμέρους εξαρτημάτων του συστήματος τηλεκατεύθυνσης παρουσιάζονται στους Πίνακες 3-10, 3-11, 3-12 και 3-13.

Πίνακας 3-10. Controller/Transmitter specifications. [26]

Manufacturer	FrSky
Model	Taranis QX7 ACCESS
Dimension	202.2mm*189.4mm*96mm
Weight	613g (without battery)
Number of channels	16 (ACCST D16) / 24 (ACCESS) channels
Internal RF module	ISRM-N
Operating Voltage Range	6.5V~8.4V
Operating Current	160mA@7.2V typ
Operating Temperature	-10℃~60℃ (14℉~140℉)
Backlight LCD resolution	128*64
Smart Port, Micro SD card slot, Mini USB Port and DSC Port	
Compatibility: ACCST D16 & ACCESS receivers	

Πίνακας 3-11. Transmitter module specifications. [27]

Manufacturer	FrSky
Model	R9M 2019
Vin Voltage Range	6.5V~13V

[26] Transmitter page: https://www.frsky-rc.com/product/taranis-q-x7-access/

[27] Transmitter module page: https://www.frsky-rc.com/product/r9m-2019/

External Power Supply	6.5V~13V
Telemetry Interface	Smart Port
Upgrade Interface	Smart Port
Compatibility	R9 series (ACCST/ACCESS)

Πίνακας 3-12. Επιπλέον τεχνικά χαρακτηριστικά R9M 2019.

	Non-LBT Version				Range Check
RF Power	10mW	100mW	500mW	100mW~1W (Self-adaptive)	0.001mW
Operating voltage/current	7.2V@120mA	7.2V@160mA	7.2V@270mA	7.2V@360mA	7.2V@50mA
Numbers of channel	16CH				/

	LBT Version			Range Check
RF Power	25mW	500mW	200mW	0.001mW
Operating voltage/current	7.2V@140mA	7.2V@400mA	7.2V@325mA	7.2V@50mA
Numbers of channel	16CH			/

Πίνακας 3-13. Receiver specifications. [28]

ACCESS protocol and supports OTA functions
900MHz/868MHz long-range with low latency
Enhanced and durable design
6 standard servo connectors (default PWM channel)
Switchable CH5/CH6 into S.Port/SBUS Output channels
Support S.Port / F.Port (Configurable in OpenTX / FrOS menu)
Signal redundancy function
External battery detection
Detachable Ipex1 connector antennas

Το σύστημα τηλεχειρισμού βασίζεται στο ανοικτού κώδικα λογισμικό OpenTX [54], επομένως οι είσοδοι του χειριστηρίου, όπως τα Joystick και οι διακόπτες προετοιμάστηκαν σύμφωνα με τις οδηγίες[29] της επίσημης ιστοσελίδας και εναλλακτικά από άλλους ανεπίσημους οδηγούς[30] [31] του διαδικτύου. Για επιπλέον παραμετροποιήσεις του υλικού, όπως συγχρονισμός του πομπού και δέκτη λήφθηκαν υπόψιν τα εγχειρίδια χρήσης που τα συνόδευαν στις συσκευασίες τους και από την επίσημη ιστοσελίδα υποστήριξης των προϊόντων του κατασκευαστή FrSky.

[28] Receiver page: https://www.frsky-rc.com/product/r9-sx/

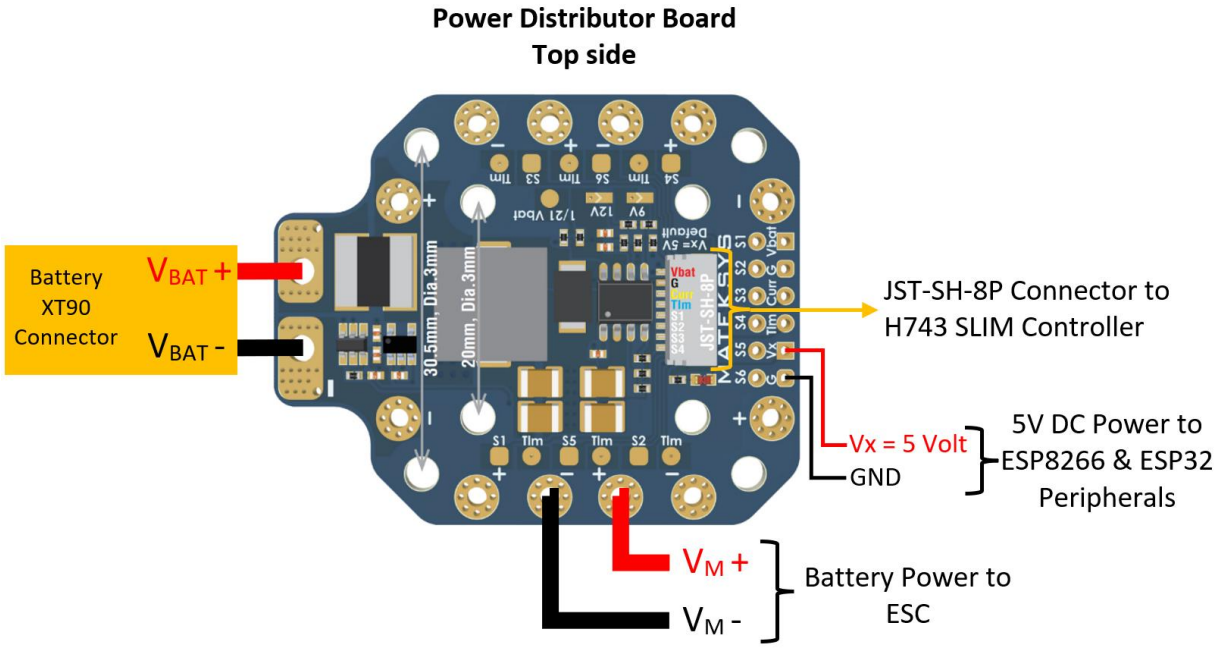
[29] OpenTX Taranis manual: https://doc.open-tx.org/opentx-taranis-manual/first-steps

[30] OpenTX input setup tutorial 1: https://oscarliang.com/setup-switch-opentx/

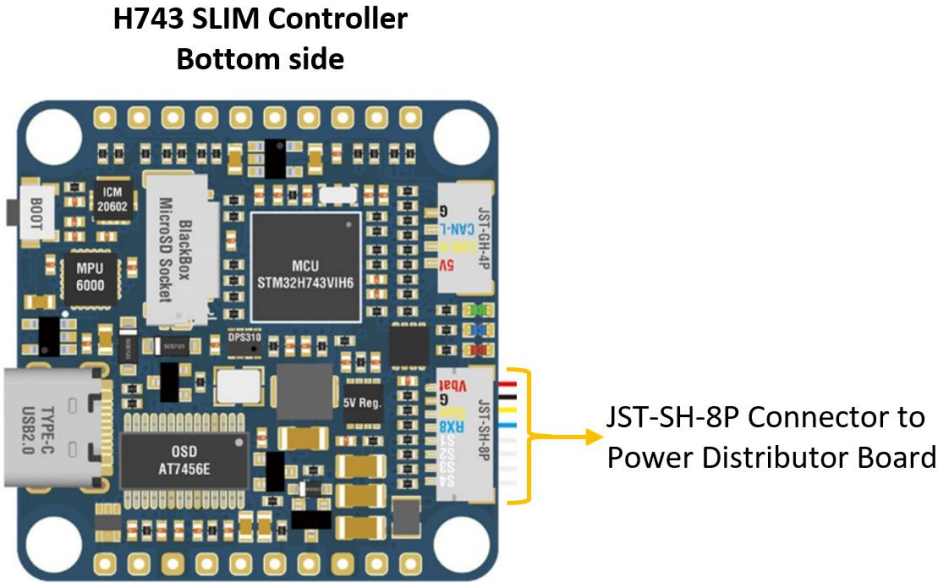
[31] OpenTX input setup tutorial 2: https://rc-soar.com/opentx/basics/index.htm

3.4.9 Διασύνδεση ηλεκτρονικών εξαρτημάτων σκάφους

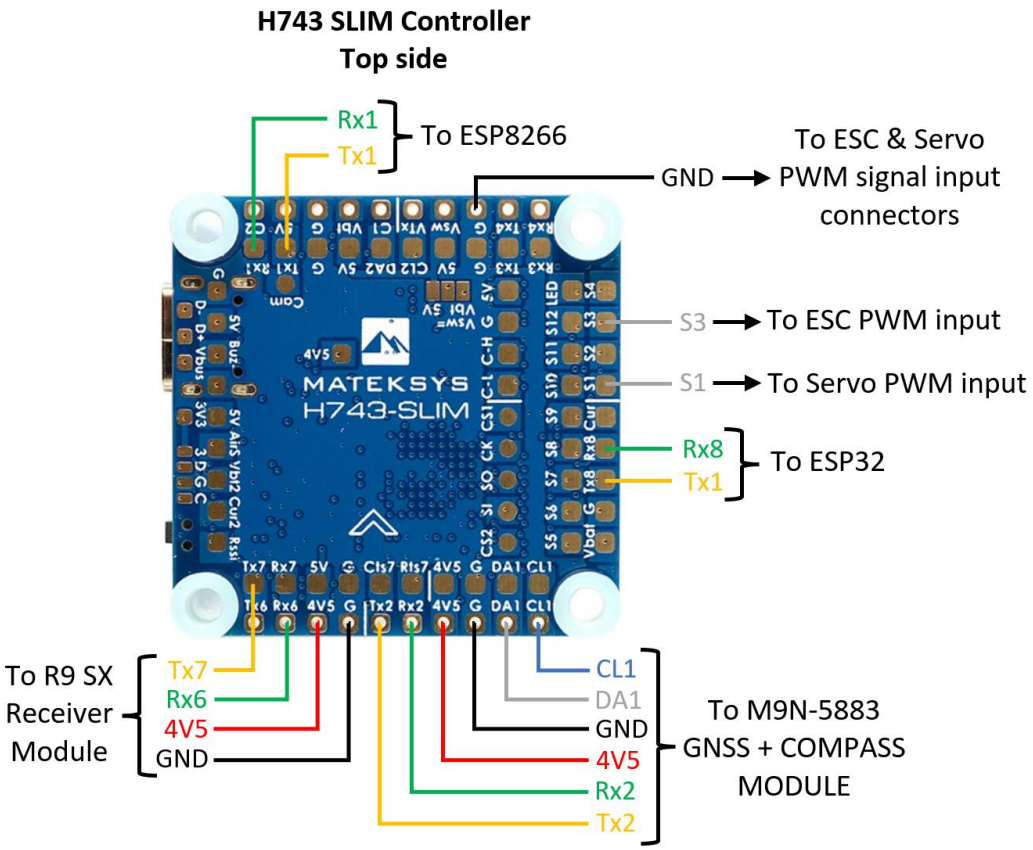
Με βάση τα φυλλάδια δεδομένων από τα ηλεκτρονικά εξαρτήματα που παρουσιάστηκαν στις προηγούμενες υποενότητες, δημιουργήθηκαν οι καλωδιώσεις διασύνδεσης των εξαρτημάτων. Επίσης για τη διεργασία αυτή, λήφθηκε υπόψιν το μηχανολογικό σχέδιο του εσωτερικού χώρου του σκάφους σε συνάρτηση με τις επιθυμητές θέσεις των ηλεκτρονικών συσκευών που θα τοποθετηθούν. Παρακάτω παρουσιάζονται αναλυτικά τα διαγράμματα διασύνδεσης και ο πίνακας τερματισμών.



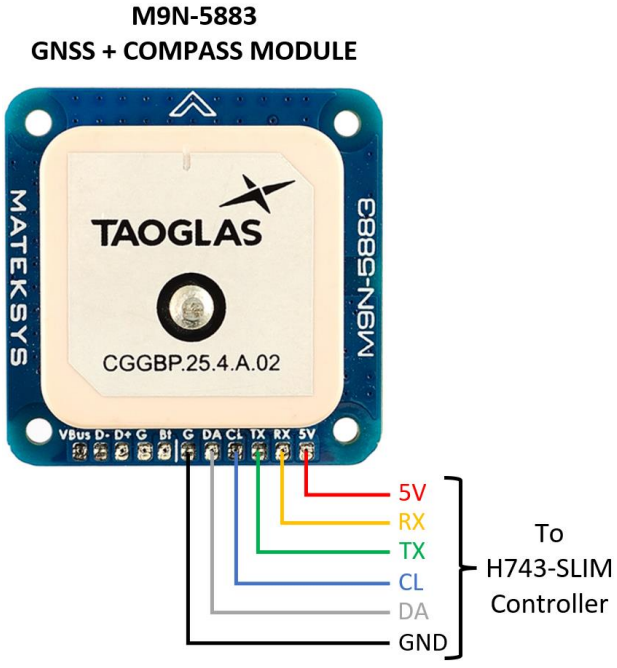
Εικόνα 3-37. Διασυνδέσεις διανομέα ισχύος



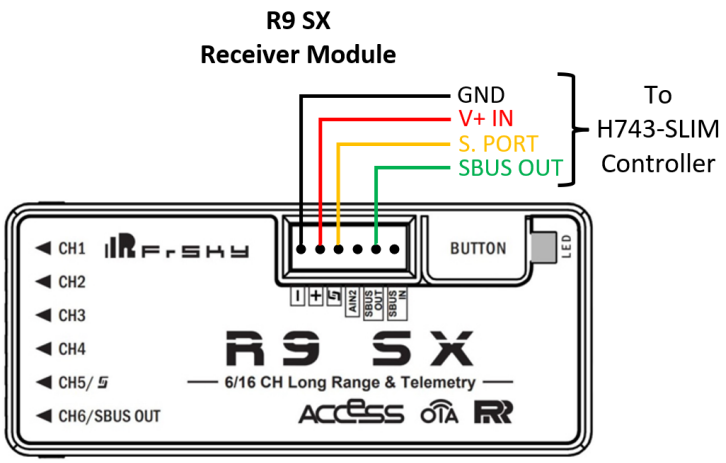
Εικόνα 3-38. Διασυνδέσεις ελεγκτή (κάτω πλευρά)



Εικόνα 3-39. Διασυνδέσεις ελεγκτή (πάνω πλευρά)

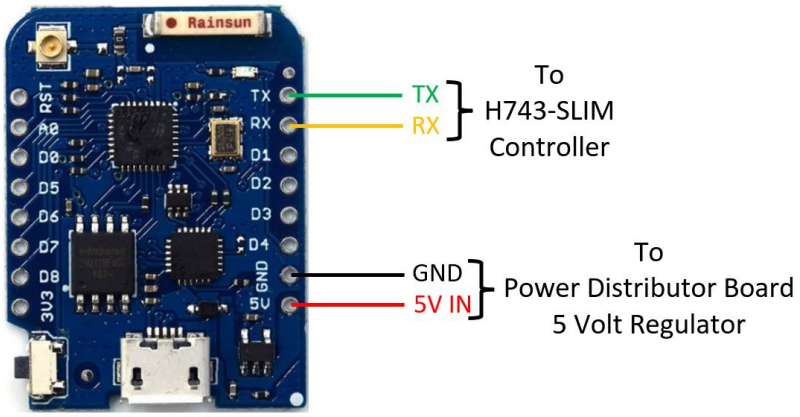


Εικόνα 3-40. Διασυνδέσεις δέκτη GNSS



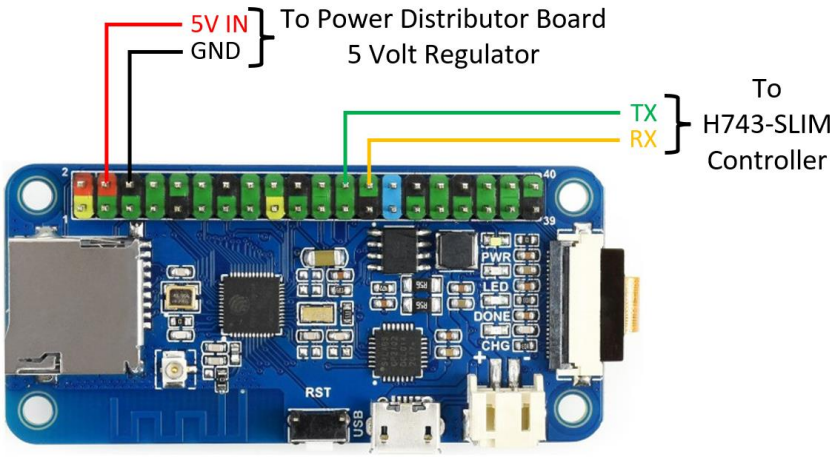
Εικόνα 3-41. Διασυνδέσεις δέκτη τηλεχειρισμού

ESP8266 WEMOS D1 MINI PRO



Εικόνα 3-42. Διασυνδέσεις αναπτυξιακού ESP8266

ESP32 ONE



Εικόνα 3-43 Διασυνδέσεις αναπτυξιακού ESP32

Πίνακας 3-14. Πίνακας τερματισμού καλωδιώσεων μεταξύ συσκευών.

Power Distributor Board	H743-SLIM
JST-SH-8P Connector	JST-SH-8P Connector
Power Distributor Board	ESP8266 WEMOS D1 MINI PRO
Vx	5V IN
GND	GND
Power Distributor Board	ESP32 ONE
Vx	5V IN
GND	GND
Power Distributor Board	ESC
V_M +	V_M +
V_M -	V_M -
Battery XT90 Connector	Power Distributor Board
V_{BAT} +	+ power input pad
V_{BAT} -	- power input pad
H743-SLIM	ESP8266 WEMOS D1 MINI PRO
Rx1	TX
Tx1	RX
H743-SLIM	ESP32 ONE
Rx8	TX
Tx8	RX
H743-SLIM	ESC
GND	GND
S3	PWM IN
H743-SLIM	Servo Actuator
GND	GND
S1	PWM IN
ESC	Servo Actuator
5V OUT	5V IN
H743-SLIM	R9 SX Receiver Module
GND	GND
4V5	V+ IN
Tx7	S. PORT
Rx6	SBUS OUT
H743-SLIM	M9N-5883
CL1	CL
DA1	DA

GND	GND
4V5	5V
Rx2	TX
Tx2	RX

3.5 Προγραμματιστικό μέρος

Στην ενότητα αυτή, θα γίνει αναφορά στην εγκατάσταση και παραμετροποίηση των λογισμικών και υπηρεσιών που χρησιμοποιήθηκαν. Επιπλέον θα παρουσιαστεί η προγραμματιστική μέθοδος που ακολουθήθηκε, για την υλοποίηση του συστήματος ελέγχου, κάνοντας χρήση των τεχνολογιών του διαδικτύου των πραγμάτων.

3.5.1 Εγκατάσταση και παραμετροποίηση Mosquitto MQTT broker

Για την επικοινωνία του σκάφους με την πλατφόρμα του Node-RED χρησιμοποιήθηκε ο MQTT broker Mosquito. Η εγκατάσταση του broker έγινε στο εικονικό μηχάνημα (Virtual Machine) που λειτουργεί στον Okeanos-knossos. Ο λόγος που επιλέχθηκε μία υπηρεσία νέφους ήταν για να δοθεί στο σύστημα δυνατότητα απομακρυσμένης διαχείρισης και παρακολούθησης μέσω διαδικτύου. Ο MQTT broker θα αναλάβει την προώθηση των μηνυμάτων κατάστασης και λειτουργίας του σκάφους από το σκάφος στην πλατφόρμα του Node-RED ενώ παράλληλα θα δέχεται και μηνύματα λειτουργίας από την διεπαφή του Node-RED και θα τα προωθεί πίσω στο σκάφος. Για την εγκατάσταση του MQTT broker ακολουθήθηκαν τα εξής βήματα:

1. Αρχικά, πρέπει να αποκτηθεί πρόσβαση στο τερματικό του Virtual Machine μέσω SSH εκτελώντας την παρακάτω εντολή στην «γραμμή εντολών» (Command promt) των Windows και εισάγεται ο μυστικός κωδικός χρήστη:

```
ssh user@83.212.77.23
```

2. Γίνεται η εγκατάσταση του Mosquitto broker με την παρακάτω εντολή:

```
sudo apt install mosquitto mosquitto-clients -y
```

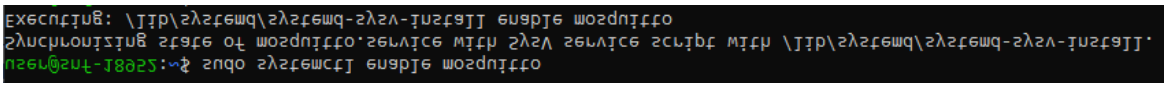
3. Αφού ολοκληρωθεί η εγκατάσταση ελέγχεται η κατάσταση του broker με την παρακάτω εντολή:

```
sudo systemctl status mosquitto
```

Εικόνα 3-44. Έλεγχος κατάστασης του broker.

4. Εκτελώντας την παρακάτω εντολή ο broker Θα εκτελείται κάθε φορά που το σύστημα του VM επανεκιννείται.

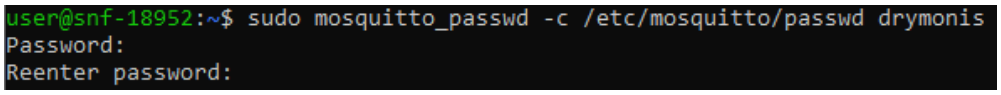
```
sudo systemctl enable mosquitto
```



Εικόνα 3-45. Εντολή αυτόματης έναρξης του broker.

5. Στη συνέχεια πρέπει να ενισχυθεί η ασφάλεια πρόσβασης στον broker με την προσθήκη ονόματος χρήστη και κρυπτογραφημένου κωδικό χρήστη. Εκτέλώντας την παρακάτω εντολή προστίθεται ένας χρήστης με όνομα *drymonis* και ζητείται η εισαγωγή ενός κωδικού χρήστη δύο φορές. Ο κωδικός αυτός θα αποθηκευτεί σε ένα αρχείο με όνομα passwd αλλά δεν θα εμφανίζεται όπως πληκτρολογήθηκε από τον χρήστη αλλά κρυπτογραφημένο.

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd Drymonis
```



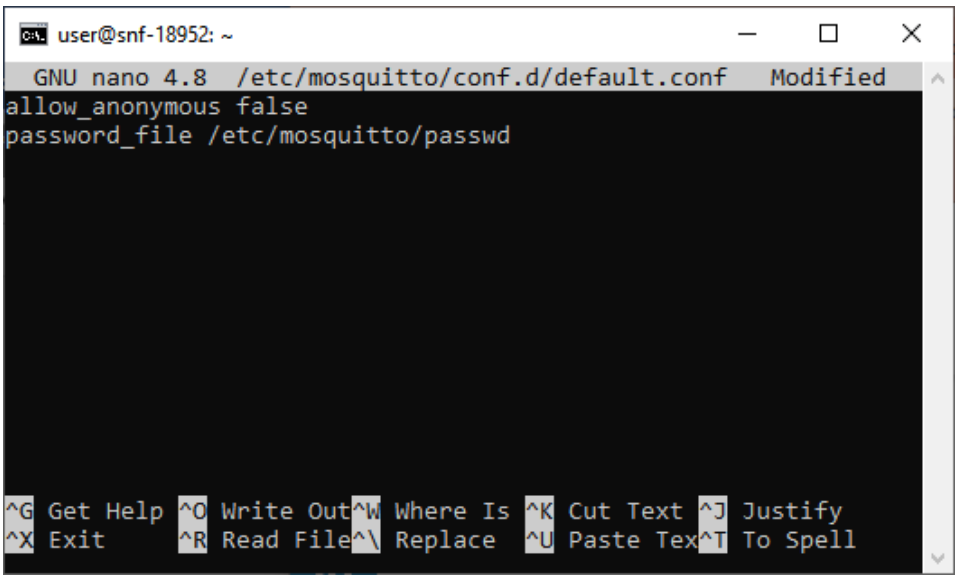
Εικόνα 3-46. Προσθήκη χρήστη και κωδικού πρόσβασης.

6. Αφού δημιουργήθηκε ο χρήστης με Username: *drymonis* και Password: *msciot20004@* πρέπει να γίνει επεξεργασία του αρχείου ρυθμίσεων του broker ώστε να μπορεί να αποκτηθεί πρόσβαση μόνο από αυτόν τον χρήστη. Για να γίνει αυτό εκτελείται η εντολή:

```
sudo nano /etc/mosquitto/conf.d/default.conf
```

7. προστίθενται οι παρακάτω γραμμές και γίνεται αποθήκευση του αρχείου:

```
allow_anonymous false
password_file /etc/mosquitto/passwd
```

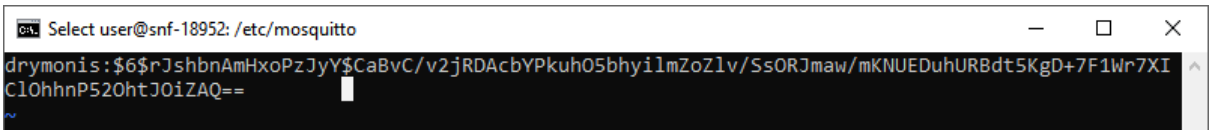


Εικόνα 3-47. Ορισμός ελεγχόμενης πρόσβασης και αρχείου κωδικών χρήστη.

8. Γίνεται επανεκκίνηση του broker και ολοκληρώνεται με αυτό τον τρόπο η εγκατάσταση και προσθήκη κωδικοποιημένης πρόσβασης σε αυτόν.

```
sudo systemctl restart mosquitto
```

Αν ανοιχθεί το αρχείο passwd εμφανίζεται ο χρήστης drymonis και ο κρυπτογραφημένος πλέον κωδικός πρόσβασης του.



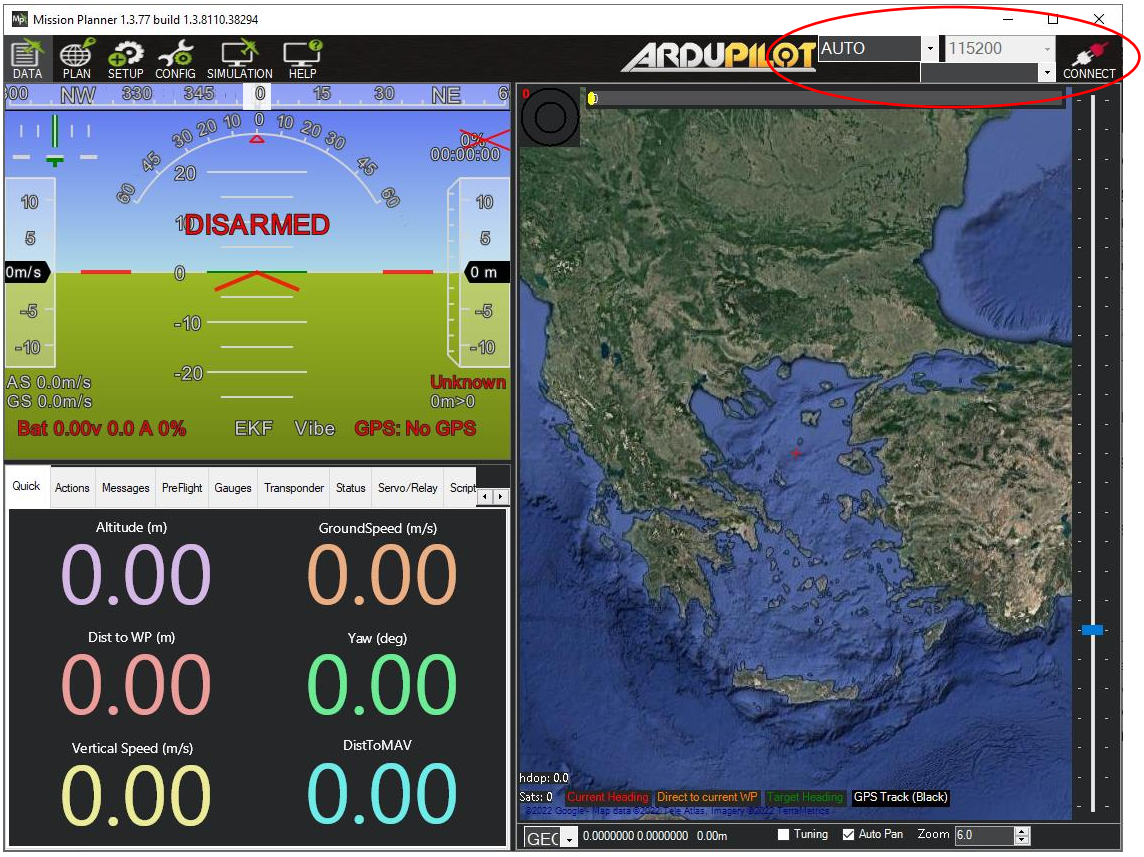
Εικόνα 3-48. Κρυπτογραφημένος κωδικός χρήστη.

3.5.2 Λογισμικό διαχείρισης Αυτόματου πιλότου Mission Planner

Το πρόγραμμα Mission Planner είναι απαραίτητο για τη παραμετροποίηση και έλεγχο της λειτουργίας του λογισμικού Ardupilot του ελεγκτή πλοήγησης H743-SLIM. Το αρχείο εγκατάστασης είναι διαθέσιμο στη νεότερη σταθερή έκδοση, από τον σύνδεσμο[32] του διακομιστή λήψεων του Ardupilot με την ονομασία “MissionPlanner-stable.msi”. Το Mission Planner εγκαταστάθηκε σε ένα υπολογιστή με λειτουργικό σύστημα Windows, από τον οποίο θα αποκτηθεί πρόσβαση στο MAVLink WiFi Bridge (ESP8266), που έχει τοποθετηθεί στο όχημα για την απομακρυσμένη διαχείριση του ελεγκτή πλοήγησης. Βασική προϋπόθεση είναι ο υπολογιστής να διαθέτει ασύρματη κάρτα δικτύου συμβατή με το πρότυπο IEEE 802.11 και τουλάχιστον μία θύρα USB για να γίνουν οι αρχικές ρυθμίσεις όπως παρουσιάζεται παρακάτω.

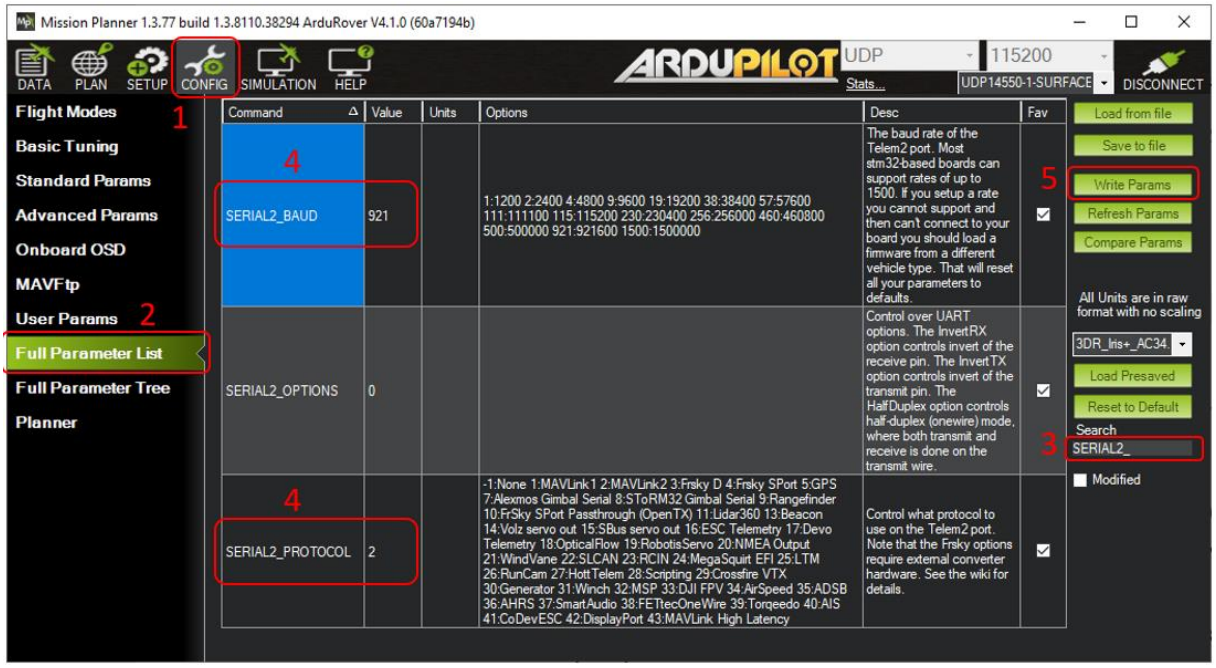
1. Μετά την εγκατάσταση του προγράμματος, έγινε η πρώτη εκκίνηση του Mission Planner και ρυθμίστηκε η πόρτα επικοινωνίας στον αυτόματο εντοπισμό όπως στην Εικόνα 3-49.

[32] Mission Planner download repository: https://firmware.ardupilot.org/Tools/MissionPlanner/



Εικόνα 3-49. Το GCS Mission Planner.

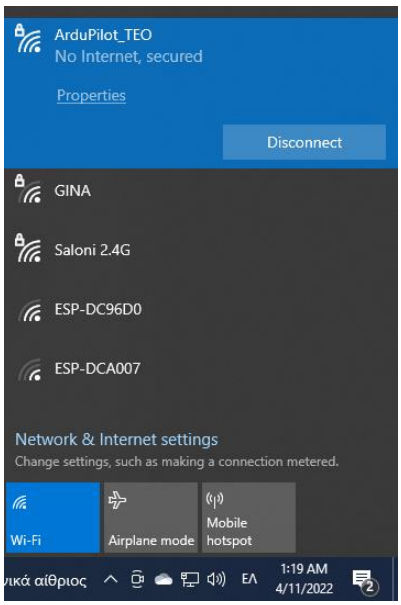
2. Συνδέθηκε η μπαταρία του οχήματος και ο ελεγκτής H743-SLIM στον υπολογιστή μέσω καλωδίου USB type A σε type C. Εφόσον οι συνδεσμολογίες ήταν σωστές θα πρέπει ενεργοποιήθηκε ο ελεγκτής και οι αντίστοιχες φωτεινές ενδείξεις LED των περιφερειακών του συσκευών.
3. Ύστερα από λίγα δευτερόλεπτα το Mission Planner συνδέθηκε αυτόματα με τον ελεγκτή, γεγονός που επιβεβαιώθηκε από τις ενδείξεις των εισερχόμενων δεδομένων που φαίνονται στη καρτέλα “DATA”.
4. Από τη καρτέλα “Config → Full Parameter List” έγινε αναζήτηση των παραμέτρων της σειριακής πόρτας “SERIAL2”. Εισάχθηκαν οι τιμές SERIAL2_BAUD: 921 και SERIAL2_PROTOCOL: 2. Τέλος αποθηκεύτηκαν οι αλλαγές στον ελεγκτή με το κουμπί “Write Params”. Στην Εικόνα 3-50 παρουσιάζεται λεπτομερώς η διαδικασία ανά βήμα.



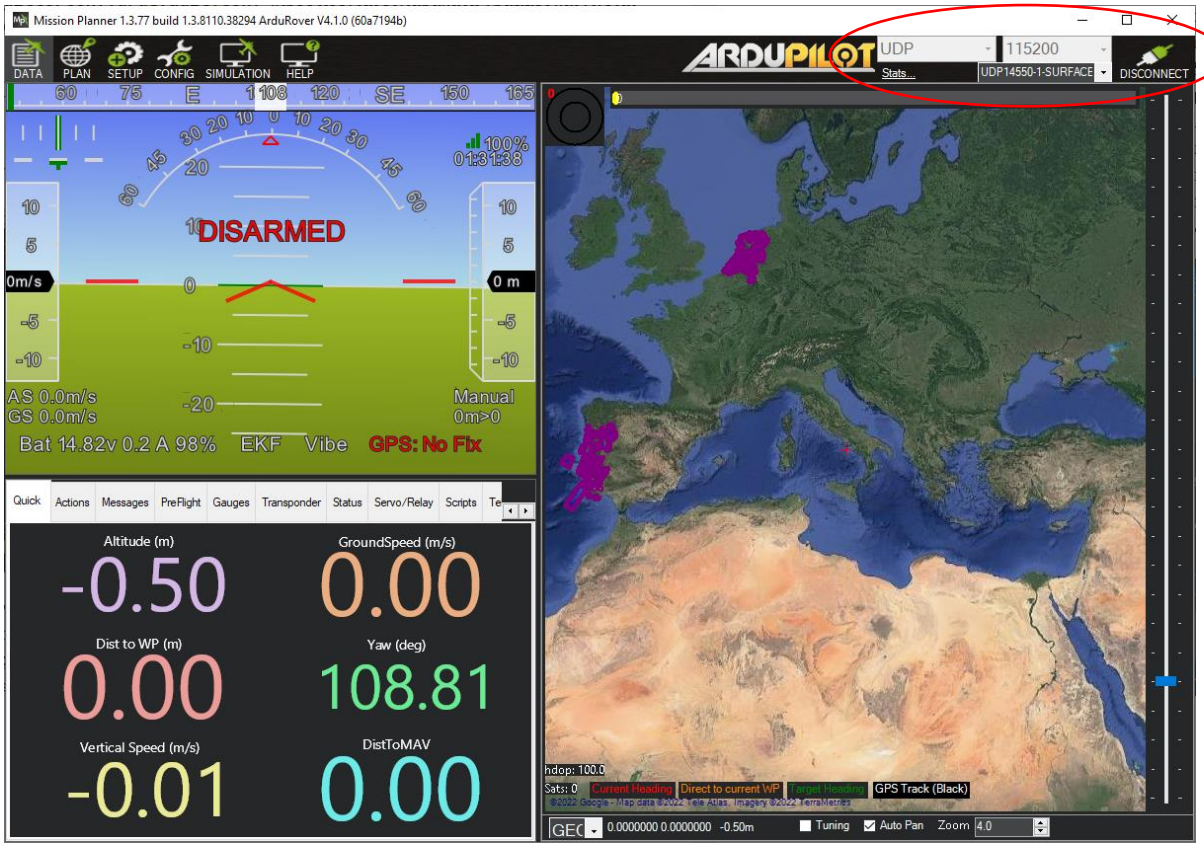
Εικόνα 3-50. Παραμετροποίηση της SERIAL2.

Έχοντας ολοκληρώσει τα παραπάνω βήματα, η επικοινωνία του H743-SLIM με το σύστημα τηλεμετρίας του ESP8266 είναι πλέον εφικτή, μέσω της ενεργοποιημένης UART πόρτας SERIAL2. Μετά από αυτή τη διαδικασία, το καλώδιο USB δεν χρειάζεται και η παραμετροποίηση μπορεί να συνεχιστεί ασύρματα μέσω WiFi. Για την ασύρματη διαχείριση μέσω Mission Planner, ακολουθήθηκαν τα παρακάτω βήματα:

1. Αποσύνδεση του προγράμματος επιλέγοντας “DISCONNECT” και αφαίρεση του καλωδίου USB.
2. Επανεκκίνηση οχήματος και Mission Planner.
3. Ύστερα από λίγα δευτερόλεπτα, έγινε ορατό από τον υπολογιστή, το SSID “Ardupilot_TEO” του σημείου πρόσβασης του ESP8266 (Εικόνα 3-51) και πραγματοποιήθηκε η σύνδεση, σύμφωνα με τα στοιχεία ταυτοποίησης που δημιουργήθηκαν στην υποενότητα 0.
4. Στη συνέχεια, το Mission Planner συνδέθηκε αυτόματα με το λογισμικό Ardupilot του οχήματος και έτσι πλέον ήταν δυνατή η απομακρυσμένη διαχείριση του ελεγκτή πλοήγησης, όπως φαίνεται στην Εικόνα 3-52. Η ένδειξη UDP (User Datagram Protocol) στην εικόνα φανερώνει την επικοινωνία μέσω του ασύρματου διαδικτυακού πρωτόκολλου WiFi.



Εικόνα 3-51. Το SSID του ESP8266.



Εικόνα 3-52. Το γραφικό περιβάλλον διαχείρισης του Mission Planner.

Για τη παραμετροποίηση της σειριακής πόρτας του MAVLINK WiFi Bridge αλλά και των υπόλοιπων περιφερειακών συσκευών του ελεγκτή πλοήγησης που χρησιμοποιούν UART, χρησιμοποιήθηκε ο Πίνακας 3-15. Ο πίνακας δημιουργήθηκε με αναφορά το φυλλάδιο

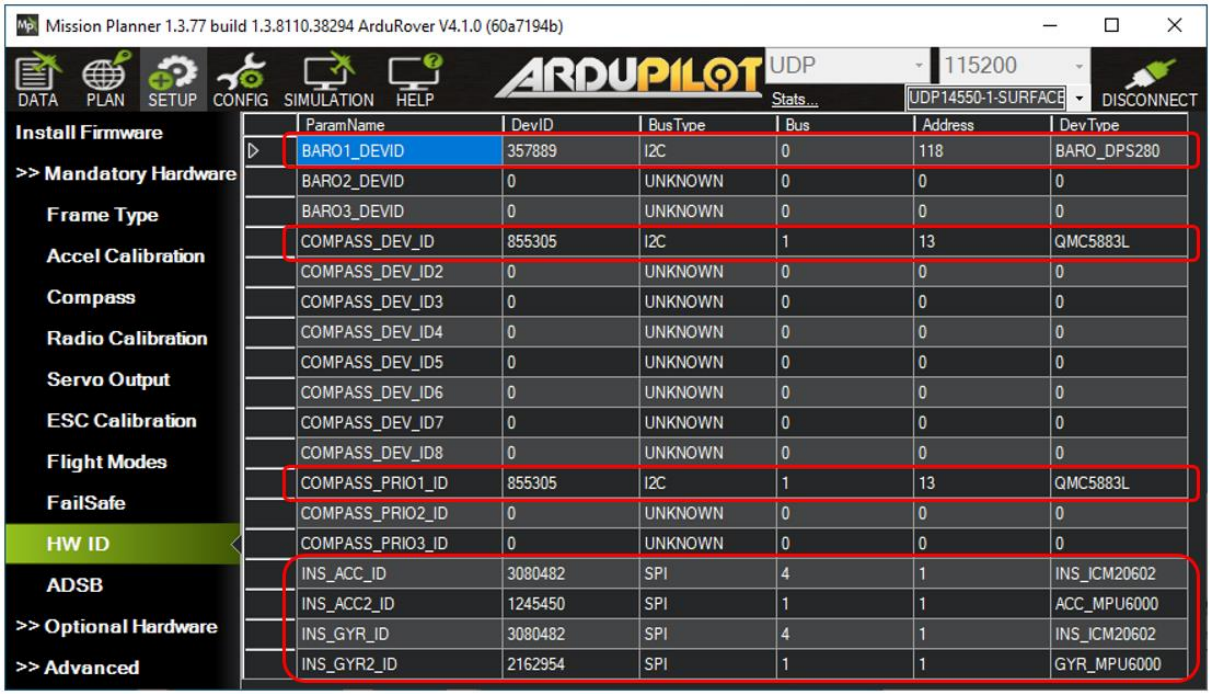
δεδομένων “Ardupilot mapping”[33] που παρέχει ο κατασκευαστής του H743-SLIM σε συνάρτηση με τις αντίστοιχες συνδεσμολογίες της υποενότητας 3.4.9.

Πίνακας 3-15. Πίνακας αντιστοίχισης των pins των σειριακών διαύλων του ελεγκτή πλοήγησης.

UART Pins	Ardupilot Serial Port Number	Function
TX7/RX7	SERIAL1	Smart Port (S. Port Receiver telemetry)
TX1/RX1	SERIAL2	ESP8266 MAVLINK WiFi Bridge (Mission Planner GCS telemetry)
TX2/RX2	SERIAL3	GNSS 1 (M9N 5883 module)
TX8/RX8	SERIAL5	ESP32 IOT Control Companion Computer
TX6/**RX6**	SERIAL7	SBUS (Receiver channel transmission)

Στη συνέχεια έγιναν οι βασικές ρυθμίσεις μεταβαίνοντας στη καρτέλα ρυθμίσεων “Setup” ακολουθώντας τα παρακάτω βήματα.

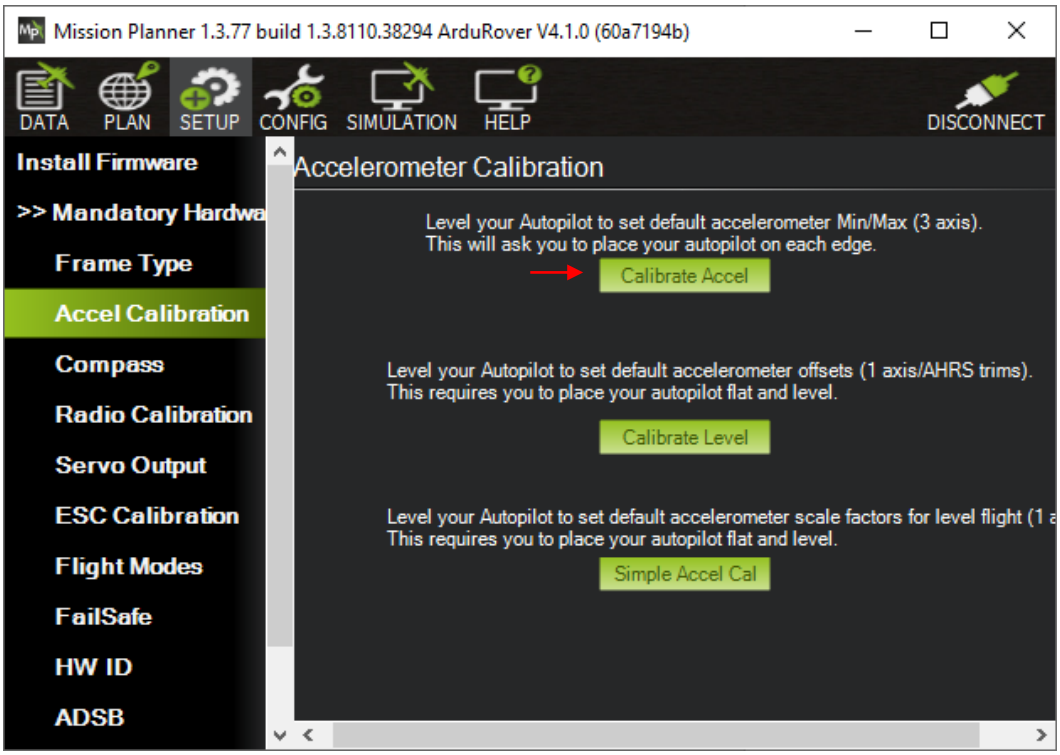
1. Επιβεβαίωση λειτουργίας των αισθητήρων IMU και του μαγνητόμετρου που είναι διασυνδεδεμένα με τον μικροελεγκτή, για τη σωστή λειτουργεία του συστήματος AHRS. Οι αισθητήρες ICM20602, MPU6000 και QMC5883L πρέπει να είναι ορατοί όταν επιλέγεται η λίστα ρυθμίσεων “SETUP → Mandatory Hardware → HW ID” (Hardware ID) όπως παρουσιάζεται στην Εικόνα 3-53. Σε αυτό το μενού παρατηρείται επίσης η ύπαρξη του αισθητήρα βαρομετρικής πίεσης DPS280, καθώς ο ελεγκτής H743-SLIM το ενσωματώνει. Παρόλα αυτά, το βαρόμετρο δεν χρειάζεται για τη πλοήγηση του σκάφους επιφανείας.



Εικόνα 3-53. Επιβεβαίωση λειτουργίας αισθητήρων πλοήγησης.

[33] H743-SLIM Ardupilot port mapping: http://www.mateksys.com/?portfolio=h743-slim#tab-id-5

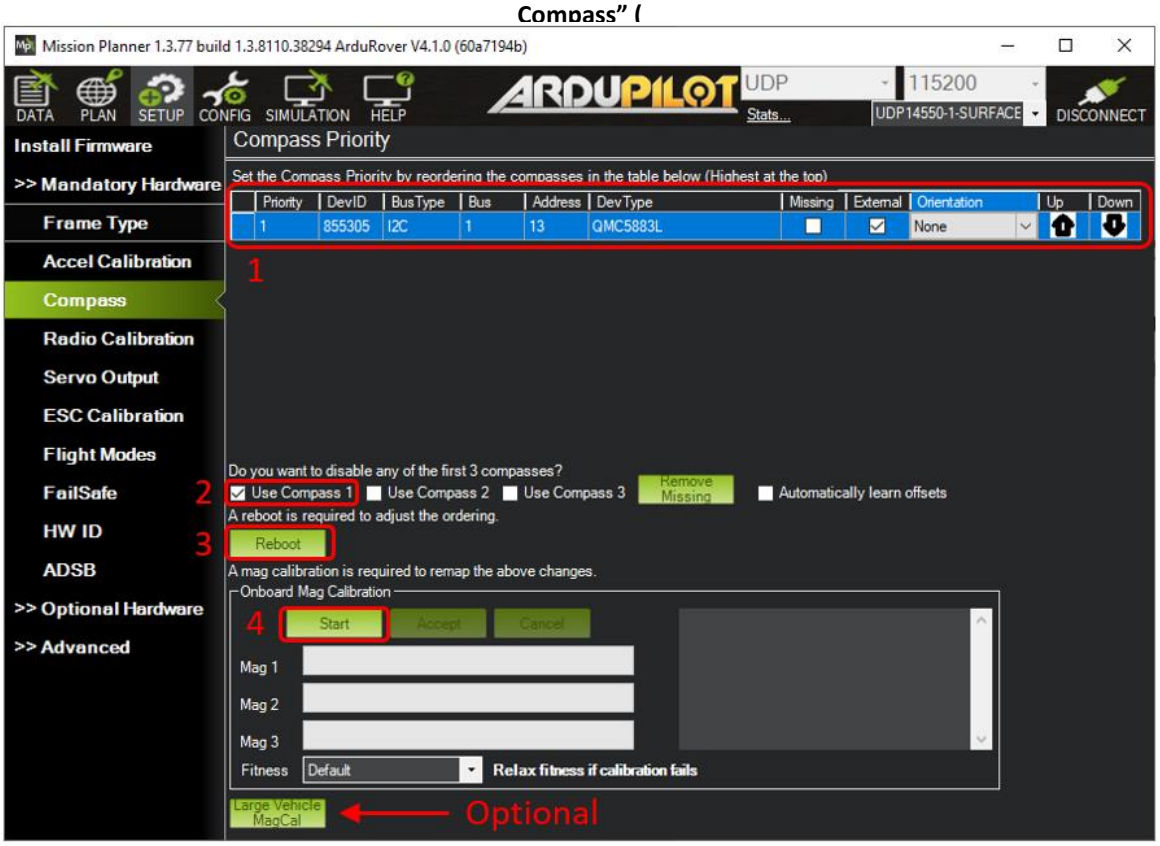
2. Στη συνέχεια έγινε βαθμονόμηση των αισθητήρων επιτάχυνσης από την επιλογή “SETUP → Mandatory Hardware → Accel Calibration” όπου επιλέχθηκε το κουμπί “Calibrate Accel” όπως φαίνεται στην Εικόνα 3-54. Σε αυτό το σημείο ακολουθούνται ρητά οι οδηγίες που δίνει στο χρήστη η εφαρμογή. Ο σύνδεσμος[34] του Ardupilot δίνει οδηγίες για τη διαδικασία.



Εικόνα 3-54. Βαθμονόμηση επιταχυνσιόμετρου

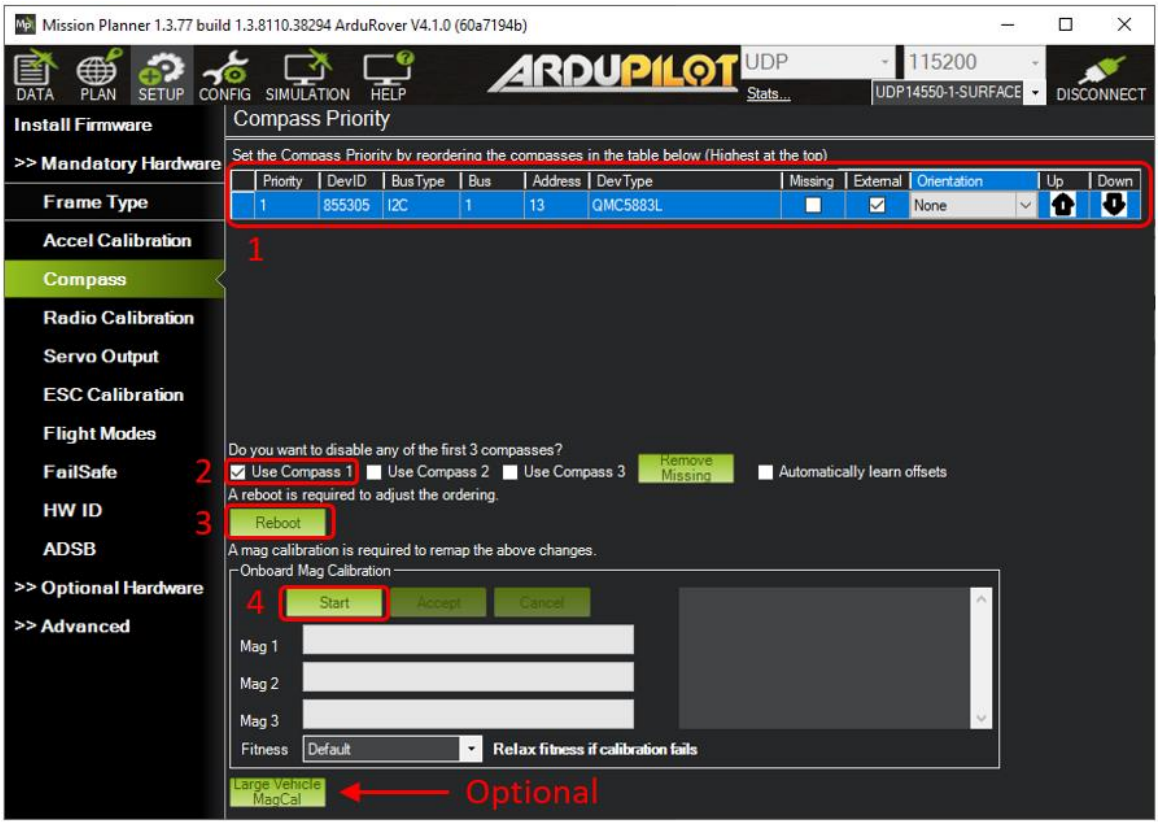
[34] Accelerometer calibration: https://ardupilot.org/copter/docs/common-accelerometer-calibration.html

Επόμενο βήμα είναι η βαθμονόμηση της πυξίδας μέσω της επιλογής "SETUP → Mandatory Hardware → Compass" (



3. Εικόνα 3-55). Σε αυτό το σημείο επιλέγεται το ενεργό μαγνητόμετρο ως πυξίδα και θα πρέπει πάλι να ακολουθηθούν οι οδηγίες της εφαρμογής. Αφότου πατηθεί το κουμπί "Start", το όχημα θα πρέπει να περιστραφεί γύρω από τον εαυτό του σε διαφορετικές γωνίες μέχρι να ολοκληρωθεί η διαδικασία. Σε περίπτωση που το όχημα είναι πολύ μεγάλο για το σηκώσει ο χρήστης, υπάρχει η επιλογή "Large Vehicle MagCal". Επιπλέον οδηγίες παρέχει η ιστοσελίδα[35] του Ardupilot.

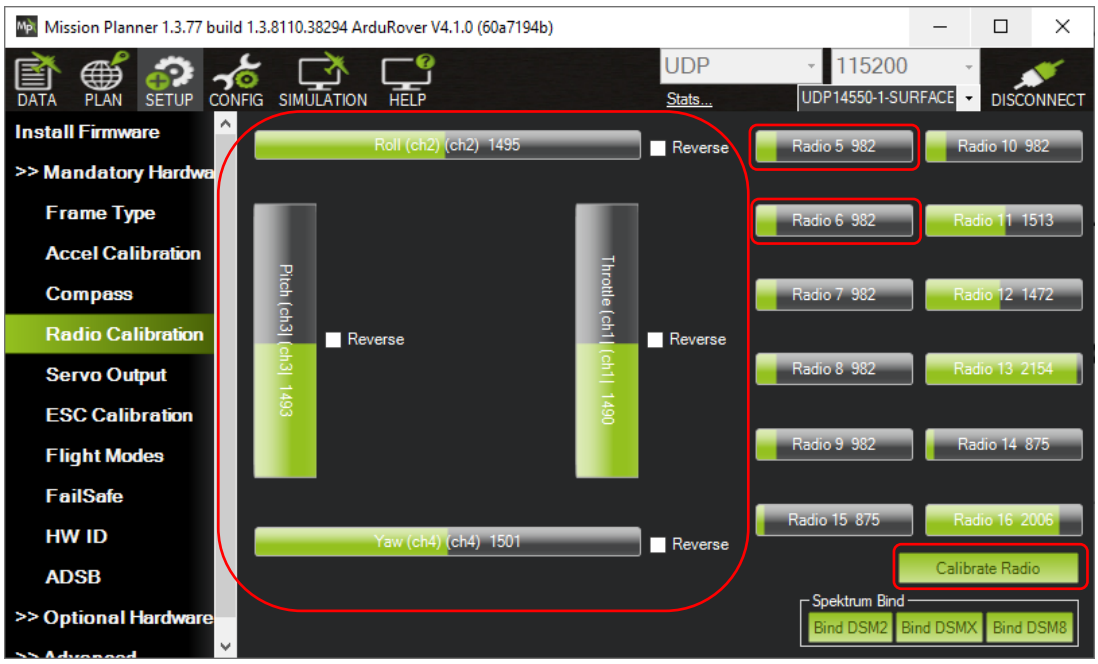
[35] Compass calibration: https://ardupilot.org/copter/docs/common-compass-calibration-in-mission-planner.html



Εικόνα 3-55. Βαθμονόμηση πυξίδας.

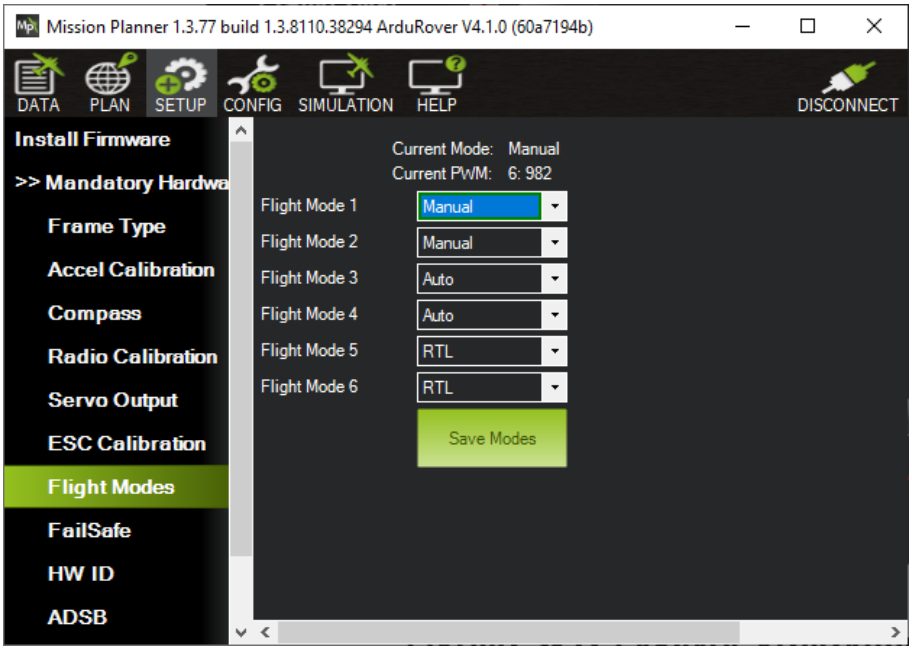
4. Επιβεβαίωση ορθής λειτουργίας των καναλιών του συστήματος τηλεχειρισμού από την επιλογή “SETUP → Mandatory Hardware → Radio Calibration” (Εικόνα 3-56). Με την προϋπόθεση ότι το τηλεχειριστήριο έχει ρυθμιστεί σωστά, πρέπει στο γραφικό περιβάλλον να αλλάζει η τιμή των καναλιών Throttle, Yaw, Roll, Pitch ανάλογα με την κίνηση των Joystick από το τηλεχειριστήριο. Επίσης πρέπει να αλλάζουν οι τιμές των καναλιών Radio5 και Radio6 όταν αλλάζει η θέση των αντίστοιχων διακοπτών του τηλεχειριστήριου. Στη συγκεκριμένη υλοποίηση, τα κανάλια που χρησιμεύουν είναι το Throttle για ώθηση, το Roll για κατεύθυνση, το Radio5 για κλείδωμα εκκίνησης και Radio6 για επιλογή λειτουργίας πλοήγησης. Σε περίπτωση που η θέση των διακοπτών και Joystick έχει απόκλιση από τις τιμές που φαίνονται στο πρόγραμμα τότε επιλέγεται το κουμπί “Calibrate Radio”. Λεπτομερής οδηγός παρέχεται από τον σύνδεσμο[36] του Ardupilot.

[36] Radio calibration guide: https://ardupilot.org/planner/docs/common-radio-control-calibration.html



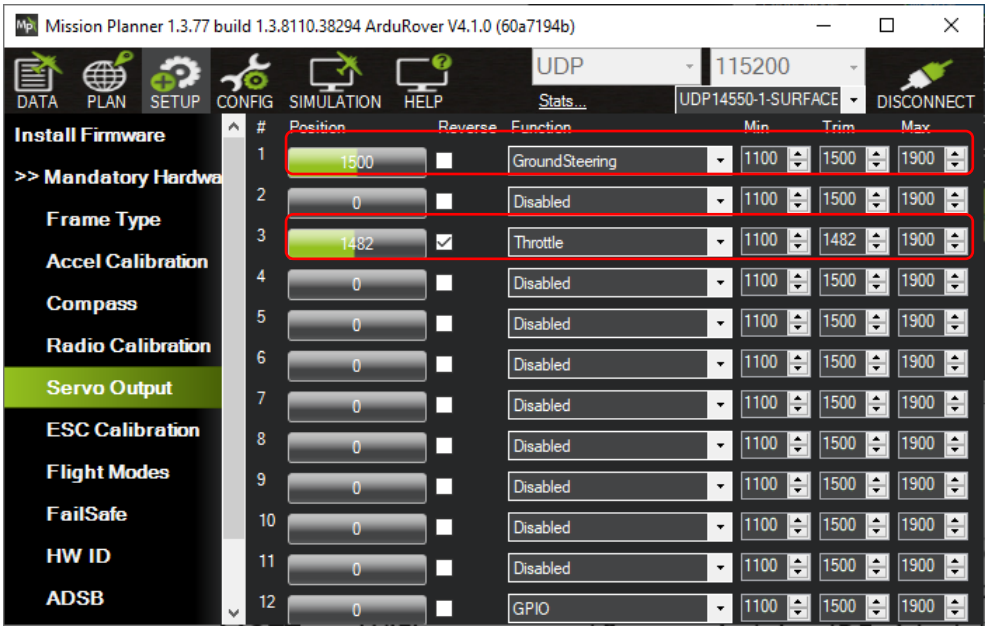
Εικόνα 3-56. Έλεγχος καναλιών του χειριστηρίου.

5. Ρύθμιση των λειτουργιών πλοήγησης, για το διακόπτη τριών θέσεων του χειριστηρίου, που αντιστοιχεί στο κανάλι Radio6 μέσω της επιλογής "SETUP → Mandatory Hardware → Flight Modes", όπως παρουσιάζεται στην Εικόνα 3-57. Με την εναλλαγή από τη λειτουργία "Manual" σε "Auto", δίνεται η εντολή στο σκάφος επιφανείας να εκτελέσει την αποστολή που του έχει ανατεθεί από το γραφικό περιβάλλον ελέγχου του Node-RED ή του Mission Planner. Έτσι δίνεται στο χειριστή η δυνατότητα επιβεβαίωσης της αποστολής για την αποφυγή ατυχημάτων. Ανάλογα με τις ανάγκες, η εκτέλεση της αποστολής μπορεί να ακυρωθεί στρέφοντας το διακόπτη στο "Manual" ή να ζητηθεί από το όχημα να επιστρέψει στο σημείο εκκίνησης, στρέφοντας το διακόπτη στο "RTL".



Εικόνα 3-57. Παραμετροποίηση λειτουργιών πλοήγησης.

6. Σύμφωνα με το διάγραμμα διασύνδεσης της υποενότητας 3.4.9 οι έξοδοι S1 και S3 του ελεγκτή H743-SLIM οδηγούν το Servo του πηδαλίου και το κύκλωμα ελέγχου ταχύτητας ESC του κινητήρα αντίστοιχα. Επομένως, παραμετροποιήθηκαν οι λειτουργίες των συγκεκριμένων εξόδων από το μενού “SETUP → Mandatory Hardware → Servo Output”, όπως στην Εικόνα 3-58 σε συνάρτηση με τον αντίστοιχο οδηγό[37] του Ardupilot. Όπου χρειάζεται, μπορεί να αντιστραφεί η έξοδος του καναλιού ελέγχου, σε περίπτωση που ο ενεργοποιητής ή ο κινητήρας κινείται αντίστροφα από το επιθυμητό. Επίσης μπορούν να ρυθμιστούν τα όρια τιμών της κάθε εξόδου σε περίπτωση που υπάρχει απόκλιση από την επιθυμητή συμπεριφορά. Ο τύπος σήματος των εξόδων ελέγχου σύμφωνα με τις αρχικές ρυθμίσεις του Ardupilot, είναι τύπου PWM για αυτό και οι τιμές ρυθμίζονται σε microseconds που αντιστοιχούν στο πλάτος των παλμών “Duty Cycle”.



Εικόνα 3-58. Ρύθμιση εξόδων ελέγχου κινητήρα και Servo.

Σε αυτό το σημείο η πλειοψηφία των βασικών ρυθμίσεων του οχήματος ήταν έτοιμη για το επόμενο στάδιο παραμετροποιήσεων. Στη συνέχεια μέσω της επιλογής “Config → Full Parameter List” παρουσιάζεται η λίστα που περιγράφει αναλυτικά όλες τις παραμέτρους του ελεγκτή πλοήγησης. Από το μενού “Full Parameter List” ρυθμίστηκαν:

- ο τύπος του οχήματος,
- τα χαρακτηριστικά της μπαταρίας και των αισθητήρων της,
- οι σειριακές θύρες για τα περιφερειακά υποσυστήματα του H743-SLIM,
- ο δέκτης γεωδαιτικών δεδομένων GNSS του οχήματος,
- οι διαγνωστικοί έλεγχοι πριν την εκκίνηση του οχήματος,
- οι επιλογές ασφάλειας εκκίνησης(διακόπτης ασφαλείας ARM/DISARM),
- και οι μέθοδοι που θα ακολουθήσει το όχημα για την αποφυγή εμποδίων.

[37] Ardupilot servo guide: https://ardupilot.org/plane/docs/servo-functions.html

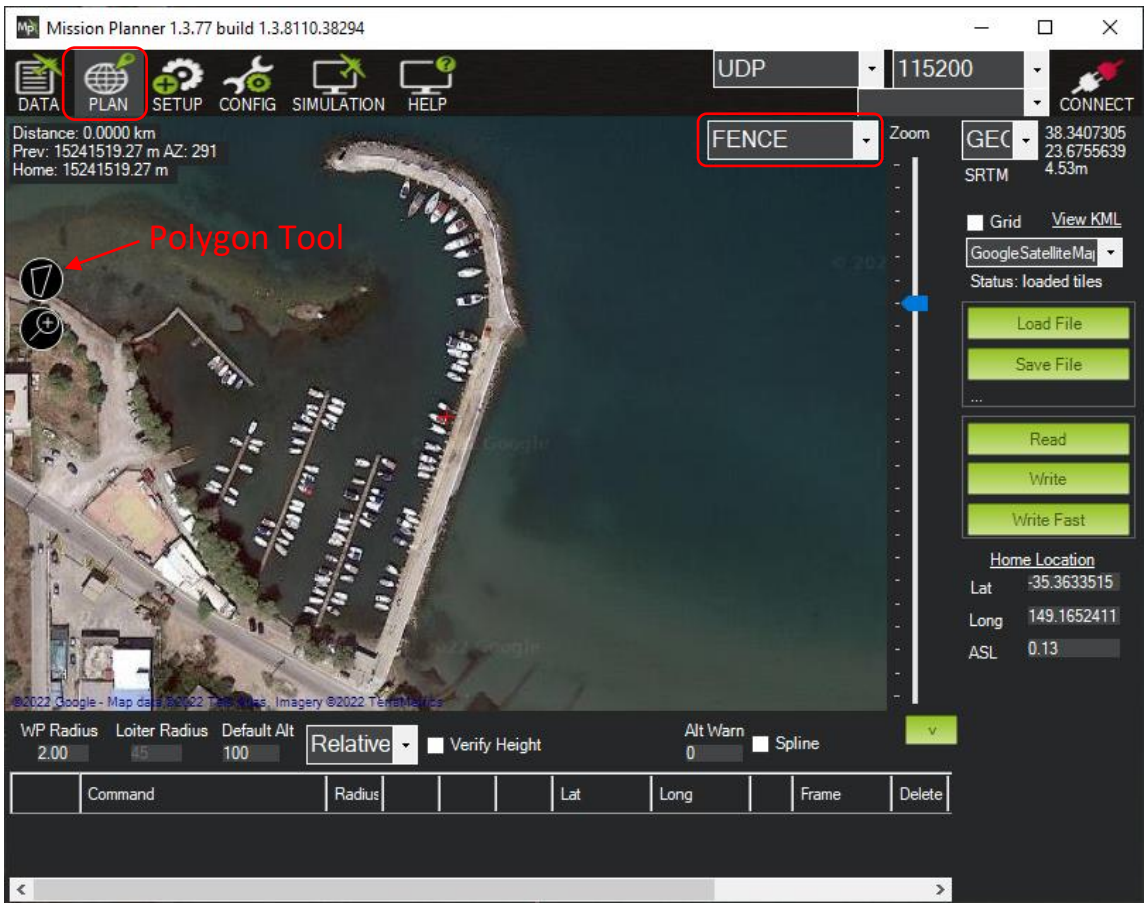
Όλοι οι παραπάνω παράμετροι μπορούν επίσης να τροποποιηθούν στα μενού “Config → Standard Params” και “Config → Advanced Params” όπου το γραφικό περιβάλλον παρουσιάζεται με πιο φιλικό τρόπο. Παρόλα αυτά, η έλλειψη εργαλείου αναζήτησης μέσα από ένα μεγάλο όγκο ρυθμίσεων, κάνει τη διεργασία ιδιαίτερα χρονοβόρα. Η λίστα των με τις ρυθμίσεις του μενού “Full Parameter List” παρουσιάζεται στο «Παράρτημα Α: Πίνακας ρυθμίσεων Ardupilot (μενού “Full Parameter List”)».

Επόμενο στάδιο, είναι η παραμετροποίηση των συστημάτων αυτομάτου ελέγχου για το πηδάλιο και την ταχύτητα του οχήματος από το μενού “Config → Basic Tuning” όπως στην Εικόνα 3-59. Στο στάδιο αυτό, πρέπει να τονιστεί ότι δεν υπάρχει κάποια συγκεκριμένη ιδανική τιμή για κάθε μία από αυτές τις επιλογές. Οι οδηγοί[38] από την ιστοσελίδα του Ardupilot και πειραματισμοί στο πεδίο δοκιμών βοήθησαν στη βελτιστοποίηση των τιμών αυτών.

Εικόνα 3-59. Μενού ρύθμισης συστημάτων κλειστού βρόγχου και πλοήγησης.

Τέλος, για ασφάλεια στο πεδίο δοκιμών και την αποφυγή στατικών εμποδίων κατά την διάρκεια της αυτόνομης πλοήγησης του σκάφους, δημιουργήθηκε ένας γεωδαιτικός χάρτης περίφραξης, γνωστός και ως “Geofence”. Στο χάρτη αυτό, ορίζεται ο χώρος λειτουργίας του οχήματος και οι περιοχές τις οποίες θα πρέπει να αποφεύγει, όπως για παράδειγμα η προβλήτα του λιμανιού, η ακτογραμμή, περιοχές επιρρεπείς στην παλίρροια και άλλων στατικών σημείων του εδάφους. Για την δημιουργία του “Geofence” επιλέγεται η καρτέλα “PLAN” όπου εμφανίζεται ο χάρτης διαχείρισης, γεωδαιτικών δεδομένων και αυτόνομων αποστολών. Στο παράθυρο υπάρχει το εικονίδιο για το εργαλείο δημιουργίας πολύγονων από συντεταγμένες που μπορεί να καθοριστεί εάν θα γίνουν φράχτες απαγόρευσης ή διέλευσης (inclusion/exclusion fence). Στην Εικόνα 3-60 φαίνεται το μενού και το εργαλείο δημιουργίας γεωδαιτικών αντικειμένων.

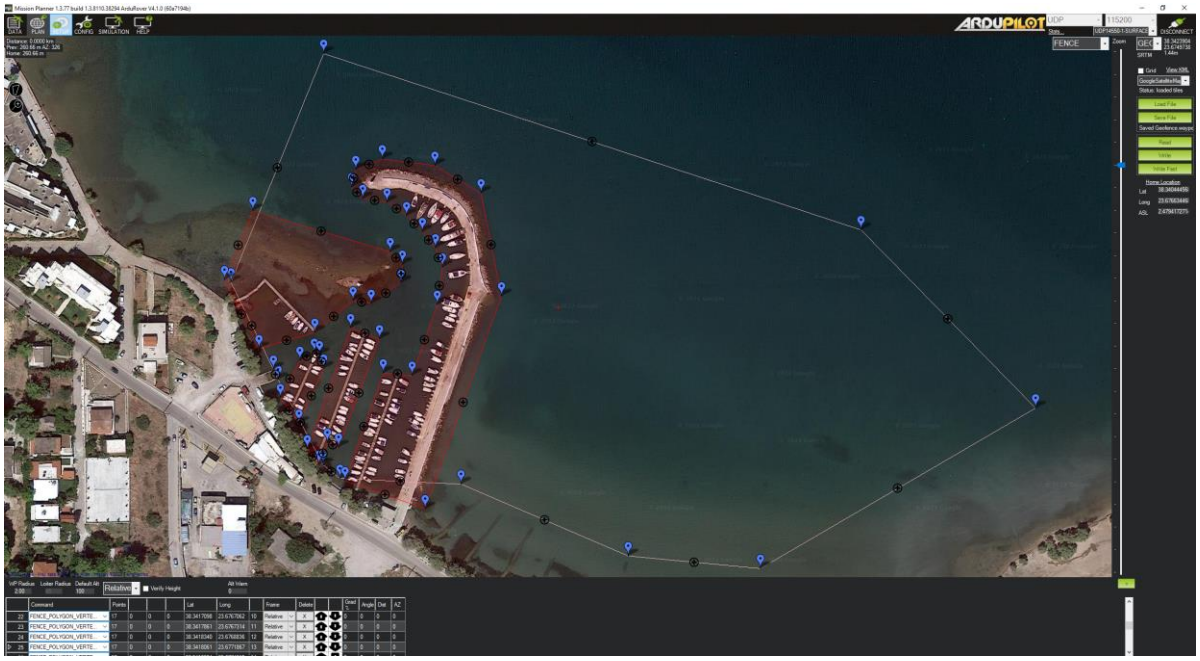
[38]Ardupilot vehicle tuning guide: https://ardupilot.org/rover/docs/rover-first-drive.html



Εικόνα 3-60. Εργαλείο δημιουργίας πολυγώνων στο γραφικό περιβάλλον.

Στο “Polygon Tool” επιλέγεται το κουμπί “Draw a Polygon” και πάνω δεξιά στο χάρτη ο τύπος δεδομένων που θα εισαχθούν, όπου στη συγκεκριμένη περίπτωση είναι φράχτης “FENCE” (Εικόνα 3-60). Στη συνέχεια με το ποντίκι επιλέγονται τα σημεία συντεταγμένων στο χάρτη, μέχρι να σχηματιστεί ο επιθυμητός φράχτης. Μόλις σχηματιστεί το πολύγωνο, επιλέγεται πάλι το εργαλείο “Polygon Tool” το οποίο θα εμφανίσει δύο νέες επιλογές, την “Fence Inclusion” και την “Fence Exclusion”, από όπου επιλέγεται ο τύπος του φράχτη. Τέλος, με την ολοκλήρωση των επιθυμητών φραχτών, επιλέγεται το κουμπί “Write” για την αποθήκευση των δεδομένων στη μνήμη του ελεγκτή πλοήγησης.

Το αποτέλεσμα της παραμετροποίησης για τη παρούσα υλοποίηση φαίνεται στην Εικόνα 3-61, η οποία αφορά τη περιοχή δοκιμών γύρω από το λιμάνι του Δηλεσίου στο νομό Βοιωτίας. Στατικά σημεία του εδάφους, όπως οι προβλήτες έχουν περιφραχτεί ως ζώνες απαγόρευσης πλοήγησης και έχουν επιτραπεί μόνο ορισμένοι διάδρομοι μέσα στο λιμάνι, από τους οποίους το σκάφος μπορεί να ξεκινήσει τη πορεία του στα ανοιχτά. Στην παρούσα υλοποίηση, το όχημα έχει ρυθμιστεί ώστε να σχεδιάζει τη πορεία που θα ακολουθήσει ανάμεσα στις γεωδαιτικές ζώνες αυτόνομα, μέσω του αλγόριθμου Dijkstra [55]. Με τη χρήση του αλγορίθμου αυτού, σκοπός είναι να περιοριστεί ο όγκος της μεταδιδόμενης πληροφορίας μεταξύ οχήματος και σταθμού ελέγχου, συμβάλλοντας στην ελάττωση του φόρτου του διαύλου επικοινωνίας.

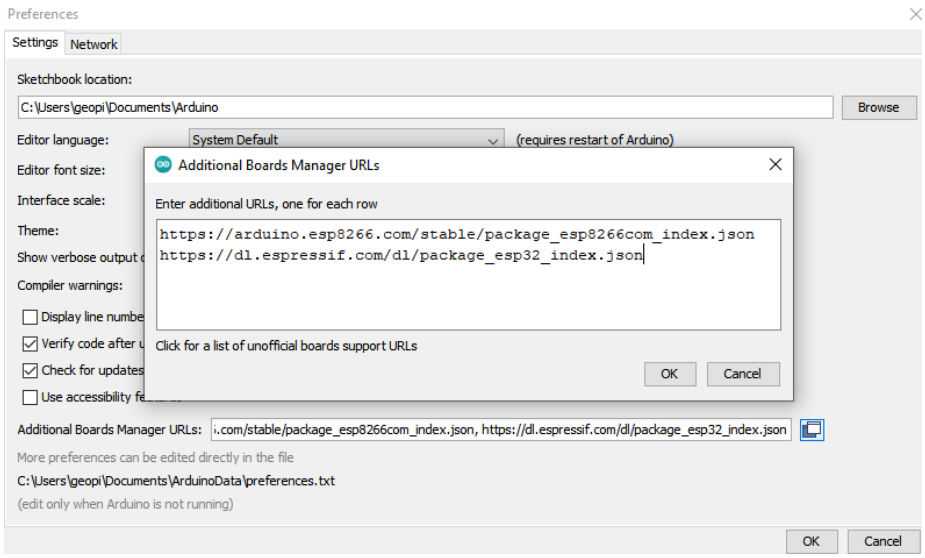


Εικόνα 3-61. Τελικός χάρτης γεωδαιτικών ορίων του σκάφους.

3.5.3 Εγκατάσταση και πρώτες ρυθμίσεις Arduino IDE

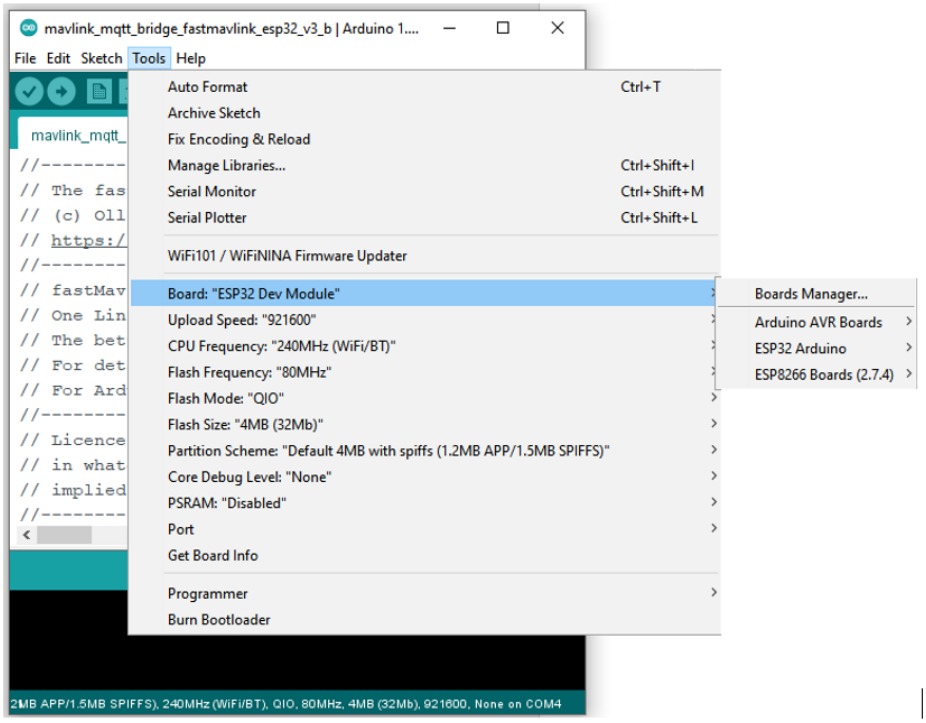
Για τον προγραμματισμό του ESP32 ώστε να επικοινωνεί με τον Flight Controller και τα πρωτόκολλα MAVLink, MQTT και WiFi χρησιμοποιήθηκε το Arduino IDE. Αφού έγινε η εγκατάσταση του μέσω του Windows Store έπρεπε να παραμετροποιηθεί κατάλληλα ώστε να μπορεί να επικοινωνεί και να αναγνωρίζει τους μικροελεγκτές ESP να εγκατασταθούν οι κατάλληλες βιβλιοθήκες πλακετών όπως επίσης και οι κατάλληλες βιβλιοθήκες για τα πρωτόκολλα επικοινωνίας. Για την εγκατάσταση των βιβλιοθηκών των μικροελεγκτών ESP ακολουθήθηκαν τα εξής βήματα:

1. Επιλέχθηκε η επιλογή Preferences από το μενού File
2. Στο νέο παράθυρο (Εικόνα 3-62), καταχωρήθηκαν οι παρακάτω διευθύνσεις URL στο πεδίο «Additional Boards Manager URLs» :https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json και επιλέγεται το «OK»



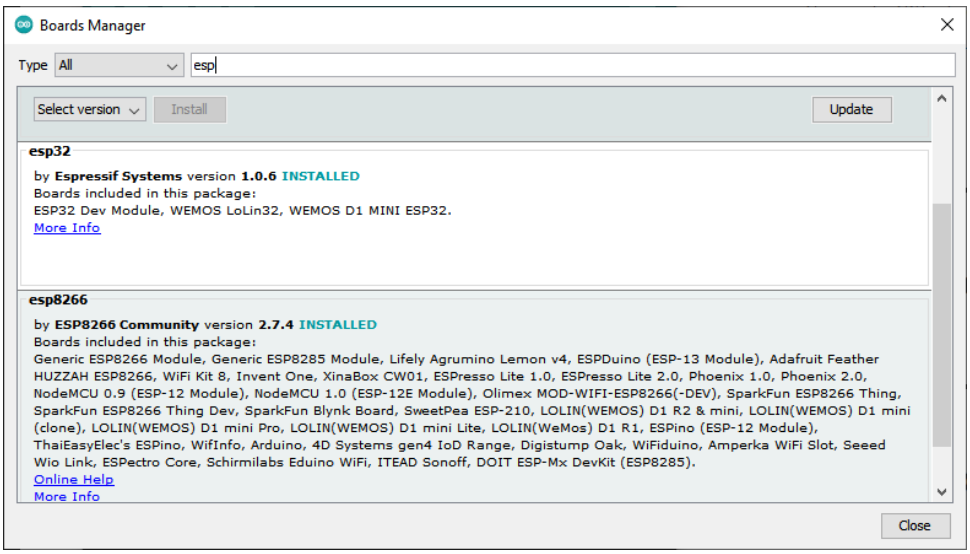
Εικόνα 3-62. Additional Boards Manager URLs.

3. Στη συνέχεια, επιλέχθηκε ο «Board Manager» (Εικόνα 3-63) από το μενού Tools>Board> Boards Manager



Εικόνα 3-63. Board Manager.

4. Στο νέο παράθυρο αναζήτησης (Εικόνα 3-64) πληκτρολογήθηκε το «ESP8266» και στην συνέχεια στο πακέτο που αφορά τα ESP8266 επιλέχθηκε το κουμπί «Εγκατάσταση». Το ίδιο έγινε αντίστοιχα και για το «ESP32»



Εικόνα 3-64. Επιλογή αναπτυξιακού από το Board Manager.

Με αντίστοιχη μέθοδο γίνεται και η εγκατάσταση των απαραίτητων βιβλιοθηκών για χρήση διαφόρων πρωτοκόλλων, αισθητήρων ή λειτουργιών που αφορούν αυτούς τους μικροελεγκτές. Αυτή την φορά όμως οι βιβλιοθήκες αναζητούνται στο μενού “Sketch → Include Library → Libraries Manager”.

3.5.4 Προγραμματισμός Companion Computer ESP32

Για τη σύνθεση του προγράμματος του μικροελεγκτή ESP32 χρησιμοποιήθηκαν δύο βιβλιοθήκες που έχουν γραφτεί στη γλώσσα προγραμματισμού C/C++ και είναι συμβατές με το περιβάλλον ανάπτυξης Arduino IDE. Η πρώτη βιβλιοθήκη, είναι η “fastMavlink”[39] και είναι υπεύθυνη για τη διαχείριση του πρωτόκολλου επικοινωνίας “MAVLink” μεταξύ των υποσυστημάτων μη επανδρωμένων οχημάτων. Η βιβλιοθήκη “fastMavlink” επιλέχθηκε για την αποδοτικότητα της σε ενσωματωμένα συστήματα με περιορισμένους υπολογιστικούς πόρους όπως μικροελεγκτές [56].

Η δεύτερη βιβλιοθήκη, είναι υπεύθυνη για τη διαχείριση του πρωτόκολλου επικοινωνίας MQTT, γνωστή με την ονομασία “arduino-mqtt”[40] και επιλέχθηκε για την υποστήριξη μετάδοσης μηνυμάτων με ποιότητα υπηρεσίας QoS2 [57]. Όπως αναφέρθηκε στην υποενότητα 2.5.2, το QoS2 εξασφαλίζει ότι τα μηνύματα θα αποσταλούν στο παραλήπτη μία μόνο φορά. Επομένως, η δυνατότητα αυτή αποτελεί σημαντικό κριτήριο για την αξιόπιστη επικοινωνία, μεταξύ διαδικτυακής πλατφόρμας υπηρεσιών και ESP32.

Το πρόγραμμα που δημιουργήθηκε παρουσιάζεται στο «Παράρτημα Β: Κώδικας Μικροελεγκτή ESP32». Ο αλγόριθμος ξεκινάει με τον ορισμό συναρτήσεων, για τη σύνθεση και αποστολή εντολών MAVLink μέσω της σειριακής θύρας UART του μικροελεγκτή ESP32, προς τον ελεγκτή πλοήγησης H743-SLIM. Επίσης ορίζεται η συνάρτηση “handleMessage” που είναι υπεύθυνη για την αναγνώριση των εισερχόμενων μηνυμάτων από τον ελεγκτή πλοήγησης και στη συνέχεια τα ανακατευθύνει στα αντίστοιχα topics του MQTT Broker.

[39] FastMavlink library github: https://github.com/olliw42/fastmavlink

[40] MQTT library github: https://github.com/256dpi/arduino-mqtt

Στον ατέρμονα βρόγχο “void loop()” του προγράμματος, διατηρείται η σύνδεση του ESP32 με το δίκτυο WiFi και κατά επέκταση ο διατηρείται ο δίαυλος ανταλλαγής πληροφοριών με τον MQTT Broker. Στη συνέχεια του ατέρμων βρόγχου, καλείται η συνάρτηση “spinOnce()”, η οποία με τη σειρά της είναι υπεύθυνη για:

- Την ανάγνωση των εισερχόμενων MAVLink μηνυμάτων από το σειριακό δίαυλο,
- τη διαχείριση τη συνάρτησης “handleMessage” καθώς και των συναρτήσεων αποστολής εντολών MAVLink όπως η “sendHeartbeat”.

Η συνάρτηση “sendHeartbeat”, καλείται ανά ένα δευτερόλεπτο για τη διατήρηση της επικοινωνίας με τον ελεγκτή πλοήγησης, ενώ οι υπόλοιπες εντολές που θα αποστείλει η “spinOnce()” καθορίζεται από τα flags και τις πληροφορίες που παρέχονται από τα αντίστοιχα MQTT topics. Για την ανάγνωση των αναφερθέντων MQTT topics από τον μικροελεγκτή, υπεύθυνη είναι η συνάρτηση “messageReceived” που καλείται έμμεσα μέσα στο βρόγχο (“void loop()”) από το αντικείμενο “client” (εντολή “client.loop();”) της βιβλιοθήκης “arduino-mqtt”. Μερικά σημαντικά τμήματα του κώδικα, παρουσιάζονται στην συνέχεια:

```
void messageReceived(String &topic, String &payload)
{
#if (enable_debug_serial == true)
  Serial.println("");
  Serial.println("incoming MQTT message: " + topic + " - " + payload);
#endif

  // Note: Do not use the client in the callback to publish, subscribe or
  // unsubscribe as it may cause deadlocks when other things arrive while
  // sending and receiving acknowledgments. Instead, change a global variable,
  // or push to a queue and handle it in the loop after calling `client.loop()`.

  if (topic == "autopilot/mission/command/arm_disarm")
  {
#if (enable_debug_serial == true)
    Serial.print("\nNew mqtt arm_disarm message: ");
#endif
    if (payload == "0")
    {
#if (enable_debug_serial == true)
      Serial.print("switching to disarmed");
#endif
      arm_disarm_switch = 0; // switch position to off
      arm_disarm_once = 1;   // enable execute once flag
    }
    else if (payload == "1")
    {
#if (enable_debug_serial == true)
      Serial.print("switching to armed");
#endif
      arm_disarm_switch = 1; // switch position to on
      arm_disarm_once = 1;   // enable execute once flag
    }
  }

  if (topic == "autopilot/mission/command/waypoint")
  {
```

Συνάρτηση ανάγνωσης εισερχόμενων MQTT μηνυμάτων. Εισάγει διεύθυνση topic και περιεχόμενο μηνύματος.

Αναγνώριση topic (arm_disarm)

Εάν το περιεχόμενο του μηνύματος είναι ο χαρακτήρας «0» (μηδέν)

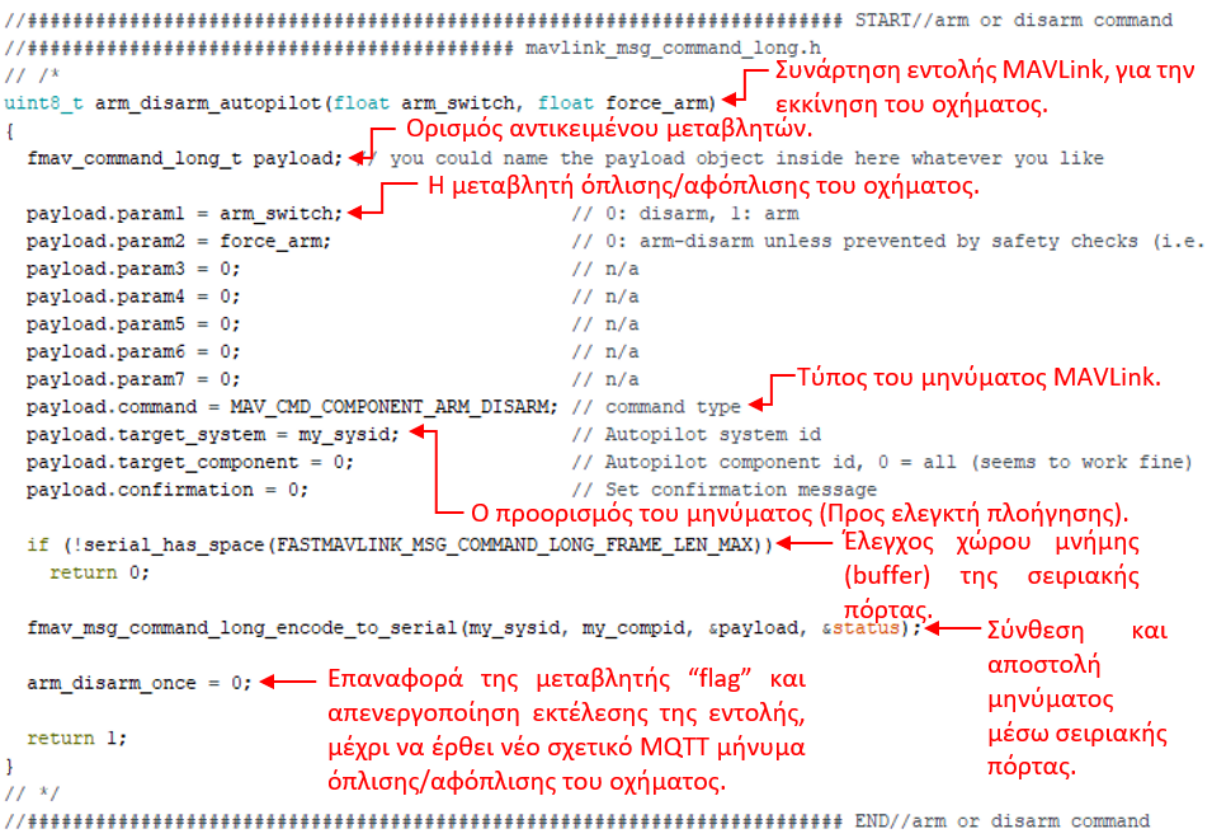
Τότε εμφάνισε μήνυμα debug και δώσε τιμές στις μεταβλητές “flags” για την εκτέλεση των αντίστοιχων συναρτήσεων αποστολής εντολών MAVLink προς τον ελεγκτή πλοήγησης.

Εάν το περιεχόμενο του μηνύματος είναι ο χαρακτήρας «1» (μηδέν)

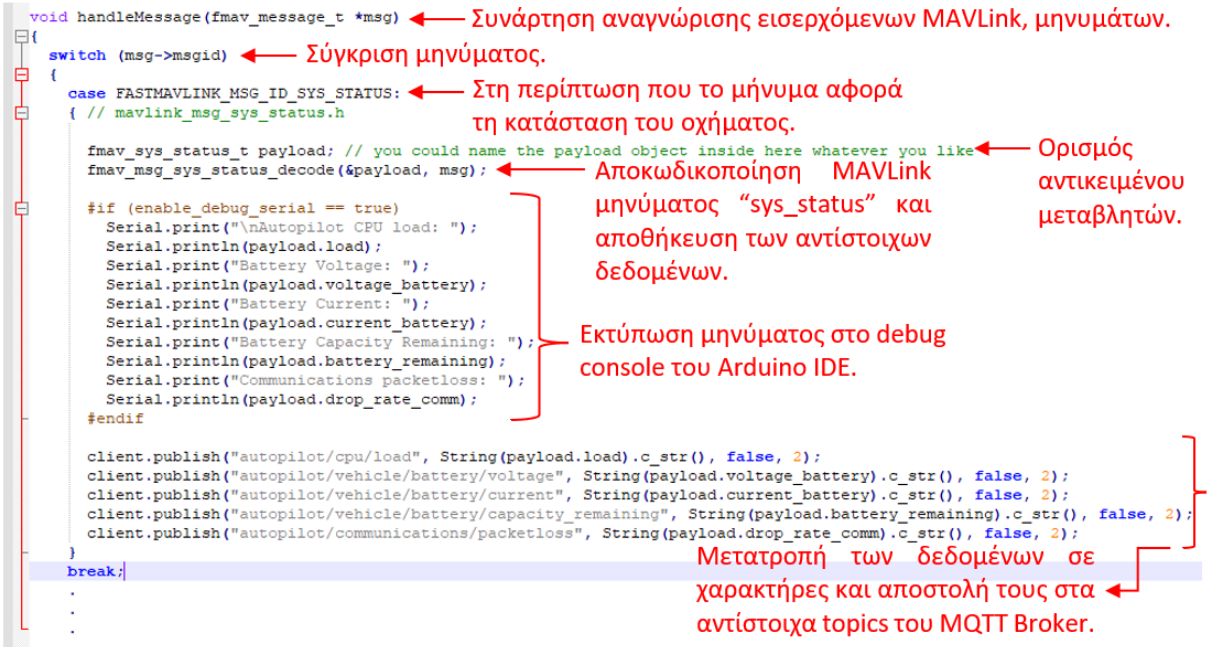
Τότε εμφάνισε μήνυμα debug και δώσε τιμές στις μεταβλητές “flags” για την εκτέλεση των αντίστοιχων συναρτήσεων αποστολής εντολών MAVLink προς τον ελεγκτή πλοήγησης.

Αναγνώριση επόμενου topic (waypoint)

Εικόνα 3-65. Σύντομη περιγραφή λειτουργίας της συνάρτησης ανάγνωσης MQTT μηνυμάτων “messageReceived”.



Εικόνα 3-66. Σύντομη περιγραφή λειτουργίας μιας συνάρτησης εντολής MAVLink.



Εικόνα 3-67. Σύντομη περιγραφή λειτουργίας της συνάρτησης εισερχόμενων MAVLink μηνυμάτων "handleMessage".

Η περιγραφή των μηνυμάτων του πρωτόκολλου MAVLink και ο τρόπος σύνταξης τους, παρουσιάζεται αναλυτικά στον σύνδεσμο[41] του οδηγού ανάπτυξης. Για την εγκατάσταση της βιβλιοθήκης “fastMavlink”, ο φάκελος “c_library” πρέπει να βρίσκεται στον ίδιο φάκελο με το αντίστοιχο αρχείο “.ino” του Arduino IDE και είναι απαραίτητο βήμα για να μεταγλωττιστεί σωστά το πρόγραμμα. Επίσης, στα αρχεία τύπου “header” του φάκελου “c_library/common” που απαρτίζουν τη βιβλιοθήκη “fastMavlink”, περιέχονται όλες οι μεταβλητές και συναρτήσεις για την κωδικοποίηση και αποκωδικοποίηση του κάθε μηνύματος MAVLink.

Στη συγκεκριμένη εφαρμογή, η χρήση του πρωτόκολλου MAVLink περιορίζεται μόνο στον σειριακό δίαυλο μεταξύ των ενσωματωμένων συστημάτων του οχήματος, ESP32 και H743-SLIM. Επομένως χρησιμοποιήθηκε η δομή επικεφαλίδας της πρώτης έκδοσης (Version 1.0 Header Structure), διευκολύνοντας την ανάπτυξη του προγράμματος.

3.5.5 Εγκατάσταση και παραμετροποίηση Node-RED

- **Εγκατάσταση Node-RED**

Το Node-RED, όπως έχει ήδη αναφερθεί, είναι μία πλατφόρμα (PaaS) που μέσω εικονοποιημένου προγραμματισμού μπορεί να διασυνδέσει διάφορες συσκευές που ίσως και να χρησιμοποιούν διαφορετικούς τρόπους επικοινωνίας. Η πλατφόρμα αυτή θα προσφέρει το γραφικό περιβάλλον διεπαφής του χρήστη με το σκάφος ενδεχομένως θα πρέπει να εγκατασταθεί σε μία συσκευή η οποία θα είναι προσβάσιμη και διαθέσιμη συνεχώς μέσω διαδικτύου. Επομένως, εγκαταστάθηκε στην εικονική μηχανή νέφους του Okeanos-knossos. Για την εγκατάσταση του ακολουθήθηκαν τα παρακάτω βήματα:

1. Αρχικά, πρέπει να αποκτηθεί πρόσβαση στο τερματικό του Virtual Machine μέσω SSH εκτελώντας την παρακάτω εντολή στην «γραμμή εντολών» (Command promt) των Windows και εισάγεται ο μυστικός κωδικός χρήστη:

```
ssh user@83.212.77.23
```

2. Γίνεται η εγκατάσταση του Node-RED με την παρακάτω εντολή:

```
sudo npm install -g --unsafe-perm node-red
```

```
user@snf-18952:~$ sudo npm install -g --unsafe-perm node-red
/usr/bin/node-red -> /usr/lib/node_modules/node-red/red.js
/usr/bin/node-red-pi -> /usr/lib/node_modules/node-red/bin/node-red-pi

> bcrypt@5.0.1 install /usr/lib/node_modules/node-red/node_modules/bcrypt
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "/usr/lib/node_modules/node-red/node_modules/bcrypt/lib/binding/napi-v3/bcrypt_lib.node" is installed
via remote
+ node-red@2.2.2
added 290 packages from 374 contributors in 18.891s
```

Εικόνα 3-68. Εγκατάσταση Node-RED.

3. Εκτελώντας την παρακάτω εντολή το Node-RED θα εκκινείται αυτόματα σε κάθε επανεκκίνηση του εικονικού μηχανήματος.

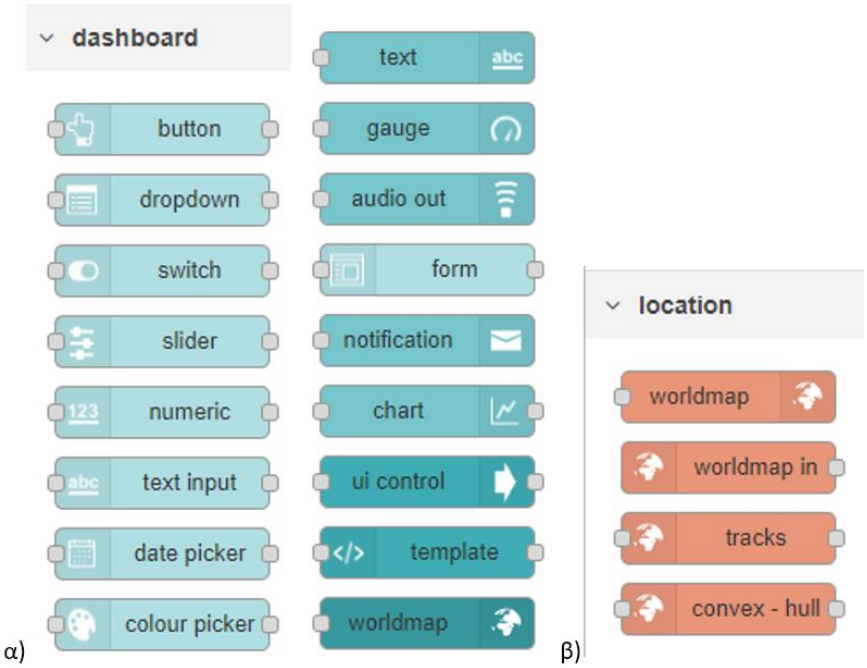
```
sudo systemctl enable nodered.service
```

4. Το Node-RED είναι πλεόν διαθέσιμο μέσω της διεύθυνσης : http://83.212.77.23:1880/

[41] MAVLink messages description: https://mavlink.io/en/messages/common.html

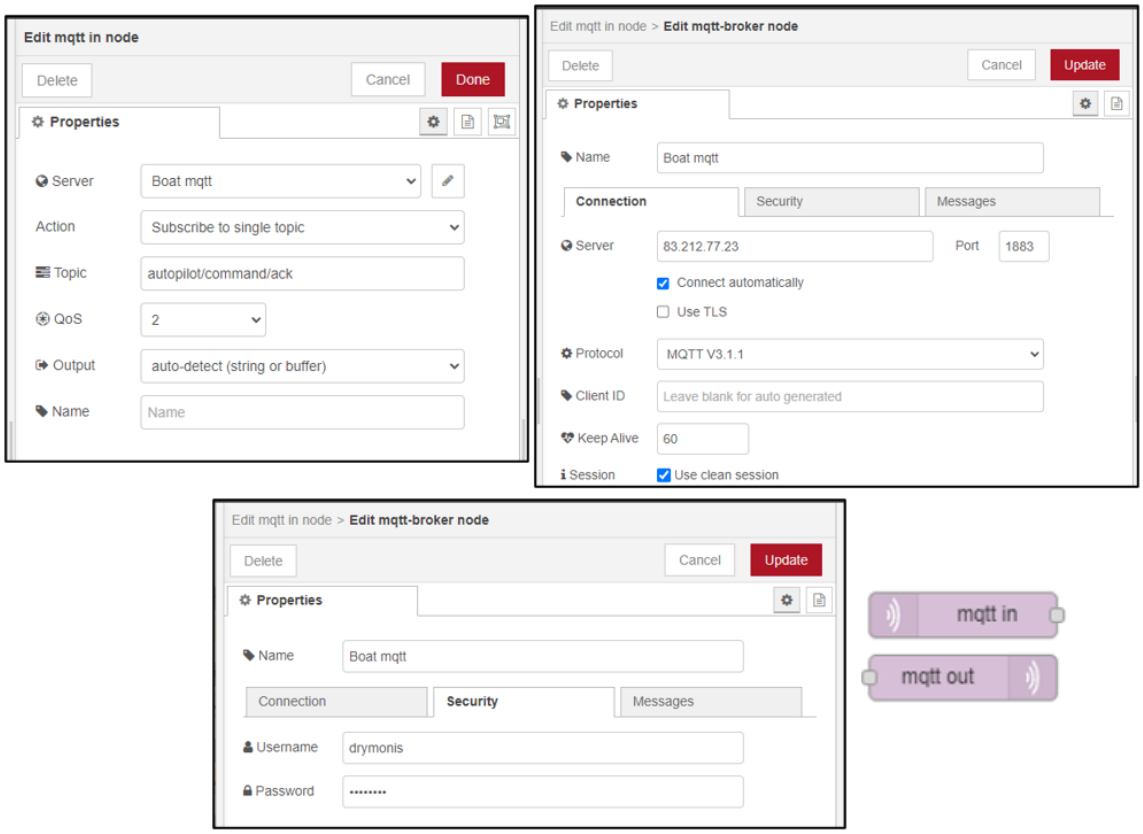
- **Προσθήκη απαραίτητων βιβλιοθηκών**

Στα πλαίσια της υλοποίησης, είναι απαραίτητη η εγκατάσταση κάποιων βιβλιοθηκών οι οποίες θα βοηθήσουν στην δημιουργία ενός κεντρικού πίνακα ελέγχου για το σκάφος. Οι βιβλιοθήκες όπως αναφέραμε και στην ενότητα "2.7 Η πλατφόρμα (PaaS) Node-RED" εισάγονται από το πλαϊνό μενού στα δεξιά επιλέγοντας το "Manage Pallete" επιλέγεται η καρτέλα "Install" και αναζητούμε τις βιβλιοθήκες *"node-red-dashboard"* και *"node-red-contrib-web-worldmap".* Η βιβλιοθήκη *"node-red-dashboard"* προσφέρει τα απαραίτητα εργαλεία διεπαφής για την δημιουργία ενός γραφικού περιβάλλοντος το οποίο θα περιλαμβάνει τις ενδείξεις δεδομένων που θα λαμβάνονται από το σκάφος. Όπως επίσης περιλαμβάνει και εργαλεία αποστολής εντολών προς το σκάφος όπως κουμπιά η πλαίσια εισαγωγής κειμένου. Τα διαθέσιμα Nodes του Dashboard είναι αυτά που φαίνονται στην Εικόνα 3-69 α). Η βιβλιοθήκη *"node-red-contrib-web-worldmap"* προσφέρει τα απαραίτητα εργαλεία ένδειξης της τοποθεσίας του σκάφους στον παγκόσμιο χάρτη. Επίσης, προσφέρει την δυνατότητα επιλογής τοποθεσίας με απλή επιλογή του σημείου απευθείας στον χάρτη. Τα σχετικά nodes φαίνονται στην Εικόνα 3-69 β).



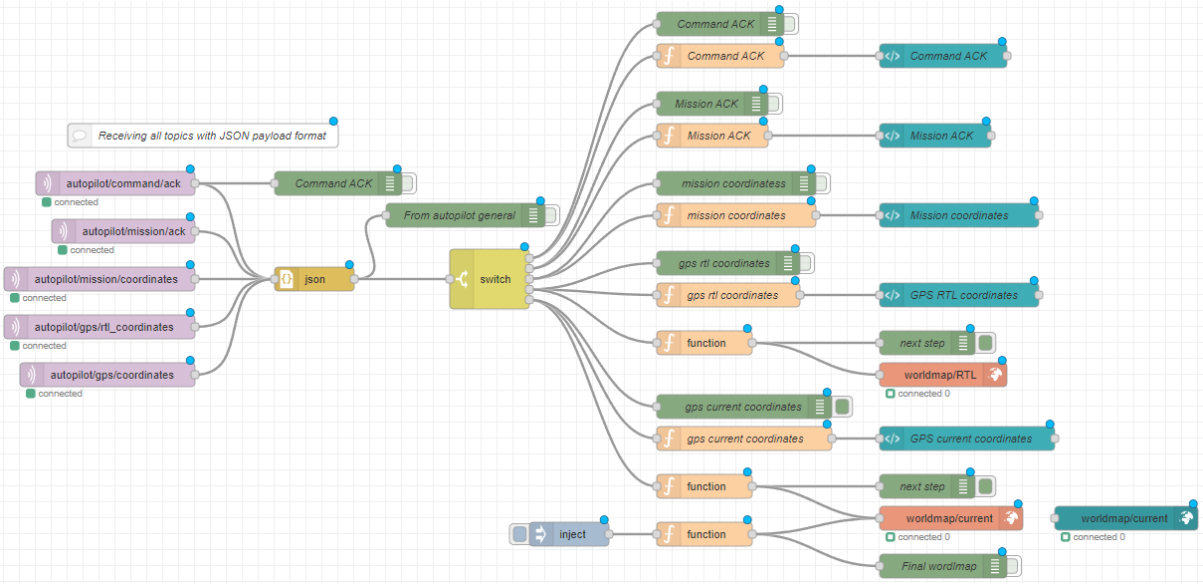
Εικόνα 3-69. Nodes για βιβλιοθήκη: α) Dashboard β) Worldmap.

Ο MQTT broker έχει ήδη δημιουργηθεί και τα topics που αφορούν τις λειτουργίες του σκάφους αλλά και τα δεδομένα πλοήγησης ή αποστολής είναι ενεργά και διαθέσιμα προς χρήση. Για να μπορέσει να εμφανίσει στο Node-RED τα μηνύματα που αποστέλλονται σε αυτά τα topics θα πρέπει να γίνει χρήση των σχετικών κόμβων "MQTT in" και "MQTT out" και να προστεθούν τα στοιχεία ταυτοποίησης του MQTT broker. Στην Εικόνα 3-70 φαίνονται οι ρυθμίσεις που έγιναν σε αυτούς του κόμβους για την λήψη και αποστολή μηνυμάτων από και προς το σκάφος.



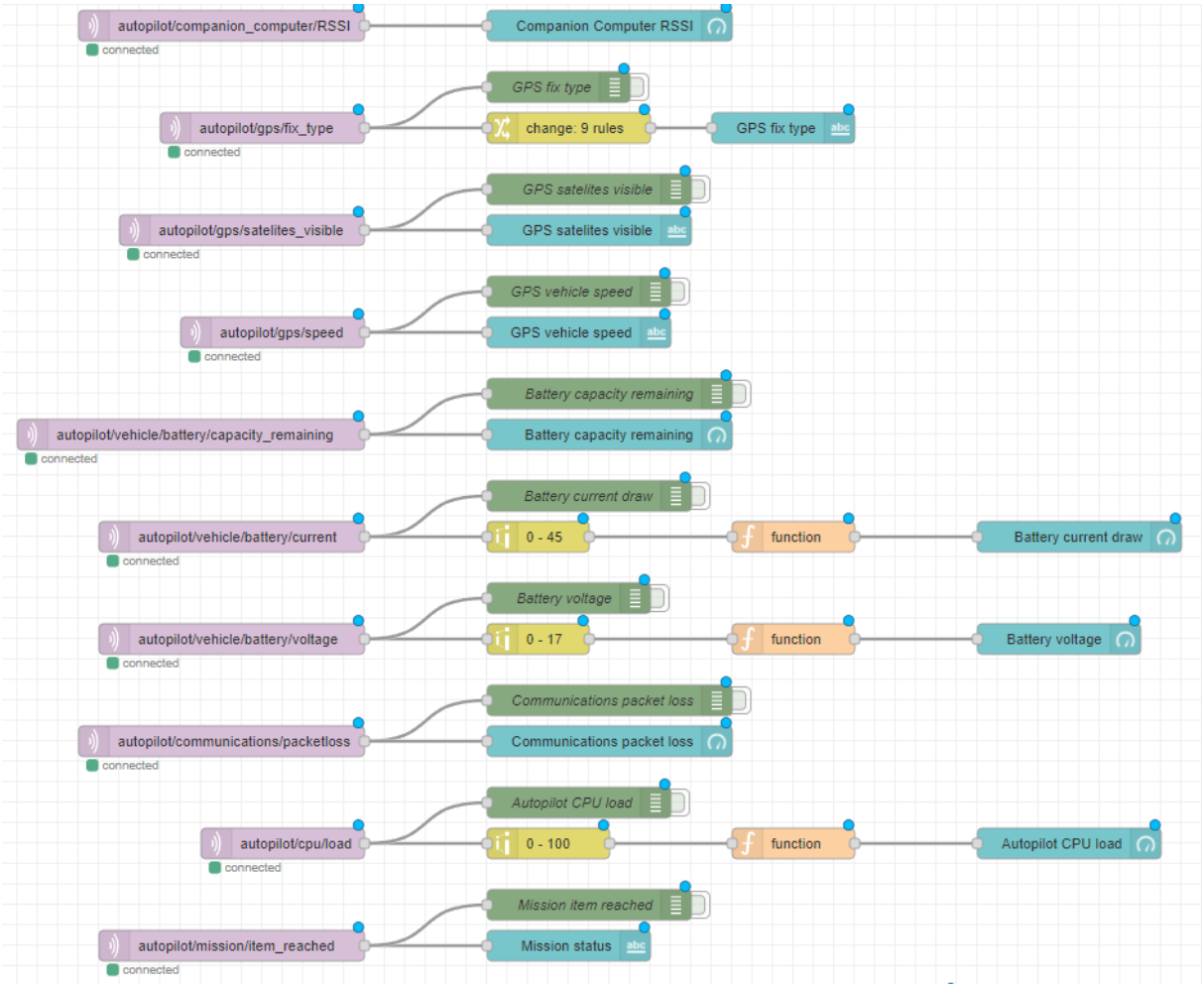
Εικόνα 3-70. Παραμετροποίηση MQTT στο Node-RED.

Στο παρακάτω διάγραμμα ροής ή αλλιώς “flow” (Εικόνα 3-71) έγινε η λήψη των μηνυμάτων όλων των topics που αφορούν δεδομένα που προέρχονται από τον αυτόματο πιλότο και αφορούν γεωγραφικές συντεταγμένες και κατάσταση λειτουργίας και εξήχθησαν στα αντίστοιχα πεδία του Dashboard (πίνακα ελέγχου σκάφους).



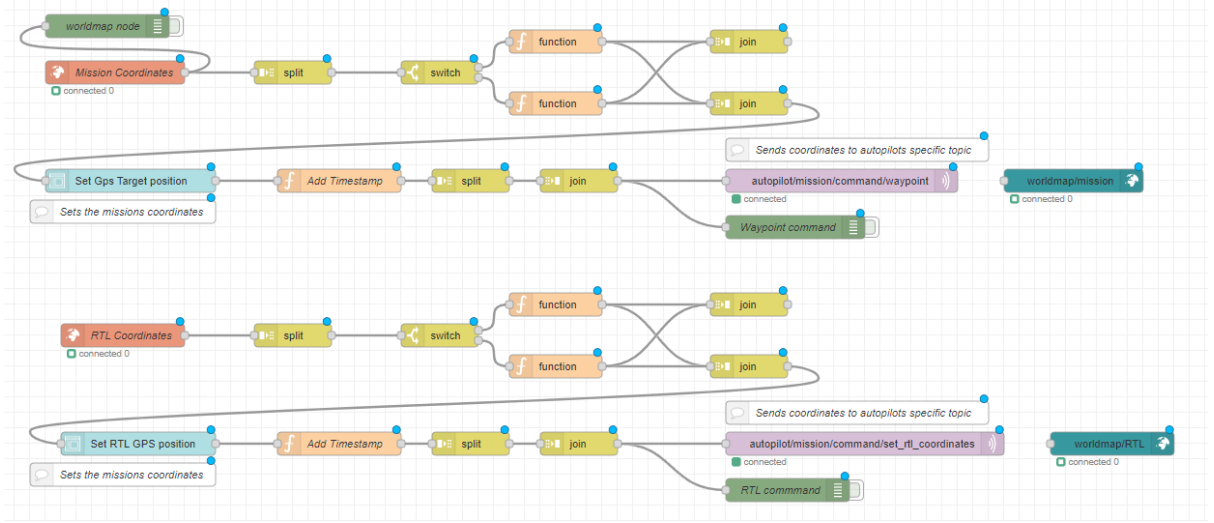
Εικόνα 3-71. Διάγραμμα ροής για την εμφάνιση των δεδομένων αποστολής.

Με τον ίδιο τρόπο αποδόθηκαν και οι υπόλοιπες πληροφορίες τηλεμετρίας του σκάφους όπως φαίνεται στην Εικόνα 3-72.



Εικόνα 3-72. Διάγραμμα ροής για την εμφάνιση των δεδομένων τηλεμετρίας.

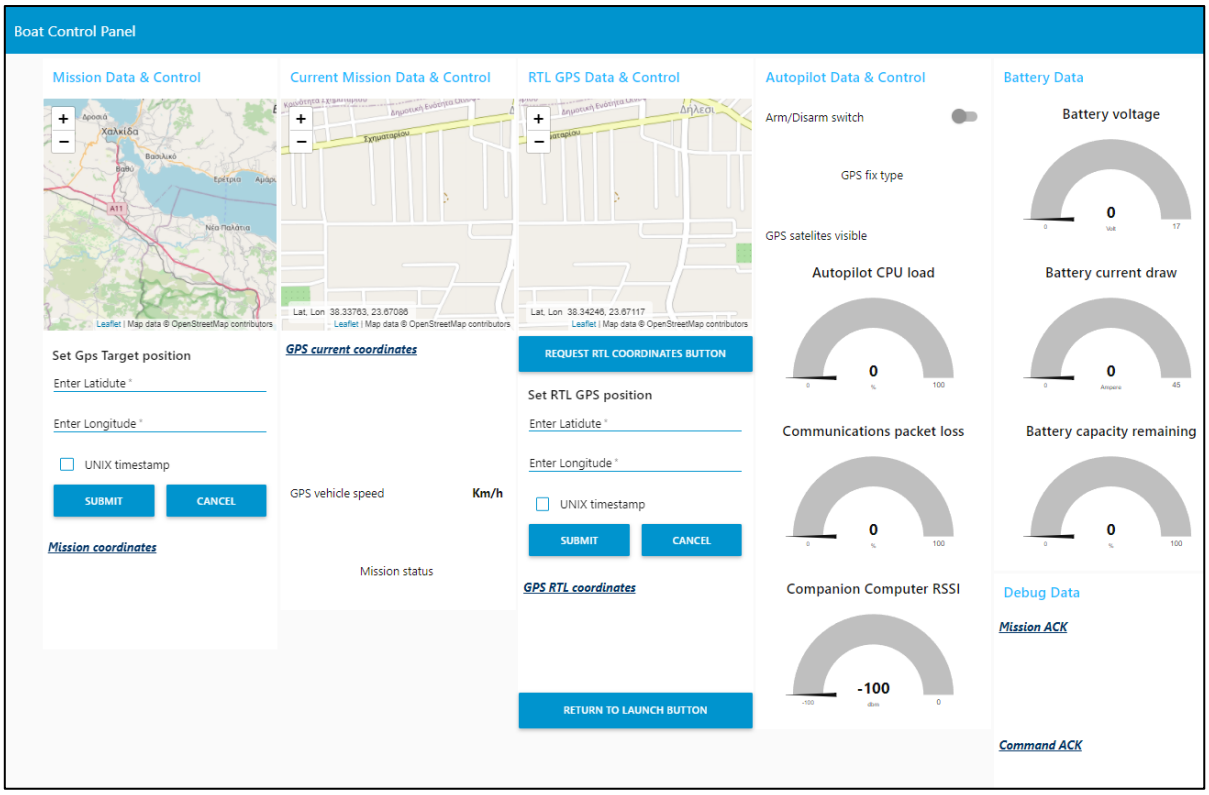
Στην Εικόνα 3-73 υλοποιήθηκε ένα διάγραμμα ροής το οποίο λαμβάνει από το dashboard δεδομένα γεωγραφικής θέσης, στη συνέχεια τα καταχωρεί στους αντίστοιχους κόμβους και μέσω MQTT μηνύματος επικοινωνεί με το σκάφος για την εκτέλεση της αποστολής.



Εικόνα 3-73. Διάγραμμα ροής για αποστολή συντεταγμένων και έναρξη αποστολής σκάφους.

Όλα τα παραπάνω διαγράμματα ροής, έχουν ως αποτέλεσμα την δημιουργία μίας διεπαφής χρήστη API στην οποία έχει πλήρη εικόνα της κατάστασης και της γεωγραφικής θέσης του

σκάφους ενώ ταυτόχρονα μπορεί να του αποστέλλει και εντολές λειτουργίας και αλλαγής αυτής της θέσης. Στην Εικόνα 3-74 φαίνεται το «Boat Control Panel - Πίνακας Ελέγχου Σκάφους».



Εικόνα 3-74 Πίνακας ελέγχου σκάφους επιφανείας - Boat Control Panel

- **Προσθήκη ελέγχου πρόσβασης**

Για την διασφάλιση της ακεραιότητας της πλατφόρμας ελέγχου του σκάφους θα πρέπει η πρόσβαση σε αυτή να είναι ελεγχόμενη. Θα πρέπει να προστεθεί έλεγχος ταυτότητας χρήστη και στο προγραμματιστικό περιβάλλον του Node-RED αλλά και στον «Πίνακα ελέγχου του σκάφους επιφανείας» . Για να γίνει αυτό ακολουθήθηκαν τα εξής βήματα:

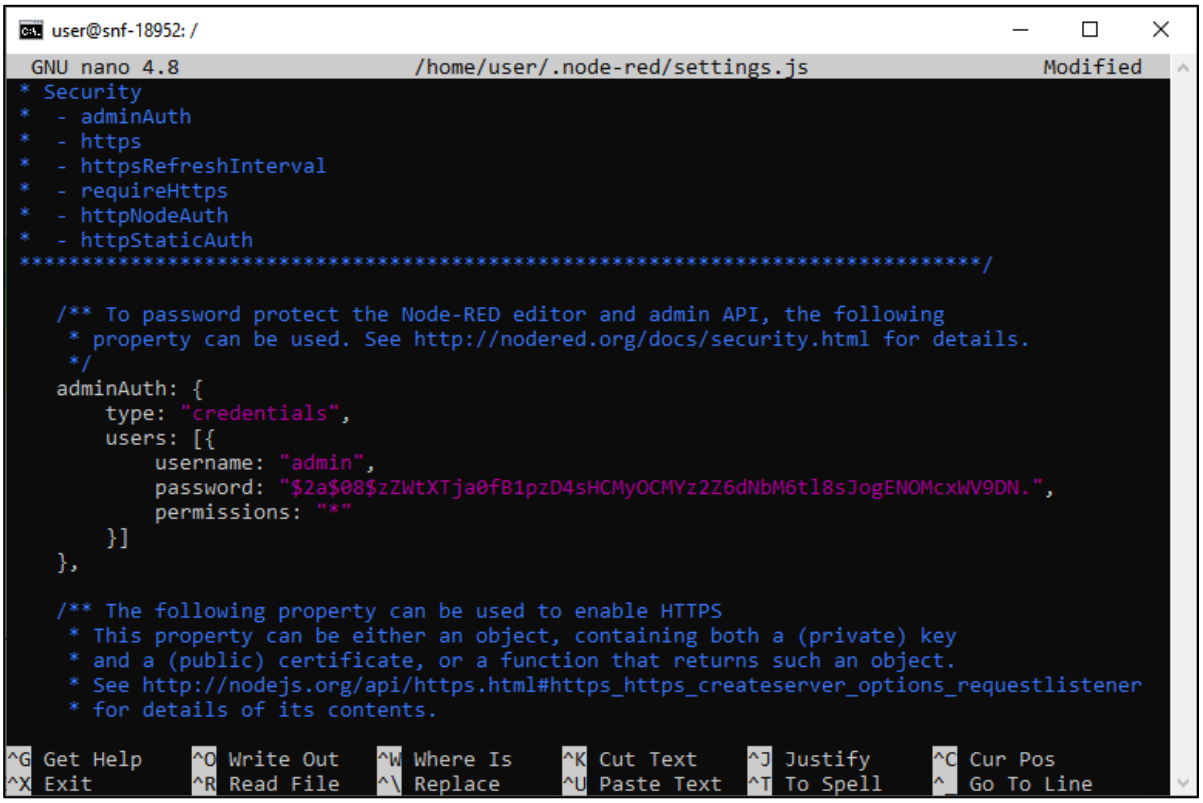
1. Αρχικά, πρέπει να αποκτηθεί πρόσβαση στο τερματικό του Virtual Machine μέσω SSH εκτελώντας την παρακάτω εντολή στην «γραμμή εντολών» (Command promt) των Windows και εισάγεται ο μυστικός κωδικός χρήστη:

```
ssh user@83.212.77.23
```

2. Στη συνέχεια, γίνεται επεξεργασία του αρχείου παραμέτρων του Node-RED με την εντολή:

```
sudo nano ~/.node-red/settings.js
```

3. Στο αρχείο που θα ανοίξει αφαιρούνται τα σχόλια (τα σχόλια έχουν σύμβολο //) από τα σημεία που φαίνονται με λευκά γράμματα στην επόμενη εικόνα (Εικόνα 3-75). Με αυτό τον τρόπο δίνεται πρόσβαση στον χρήστη admin . Το password είναι hashed, δηλαδή σε μία κρυπτογραφημένη μορφή. Αν προστεθούν και άλλοι χρήστες θα έχουν και αυτοί password σε κρυπτογραφημένη μορφή με διαφορετικό συνδυασμό συμβόλων και γραμμάτων. Με Ctrl+X και Y αποθηκεύεται και κλείνει το αρχείο κειμένου.

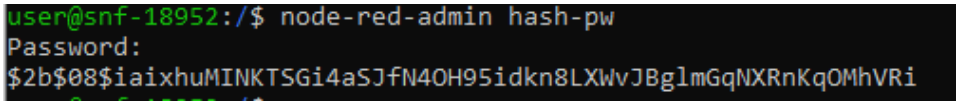


Εικόνα 3-75 Επεξεργασία αρχείου settings.js

4. Για να γίνει αλλαγή του αρχικού κωδικού χρήστη εκτελείται η παρακάτω εντολή :

`node-red-admin hash-pw`

Η εντολή αυτή θα ζητήσει ένα password και θα αποδώσει τον ίδιο κωδικό σε μορφή hash (Εικόνα 3-76).

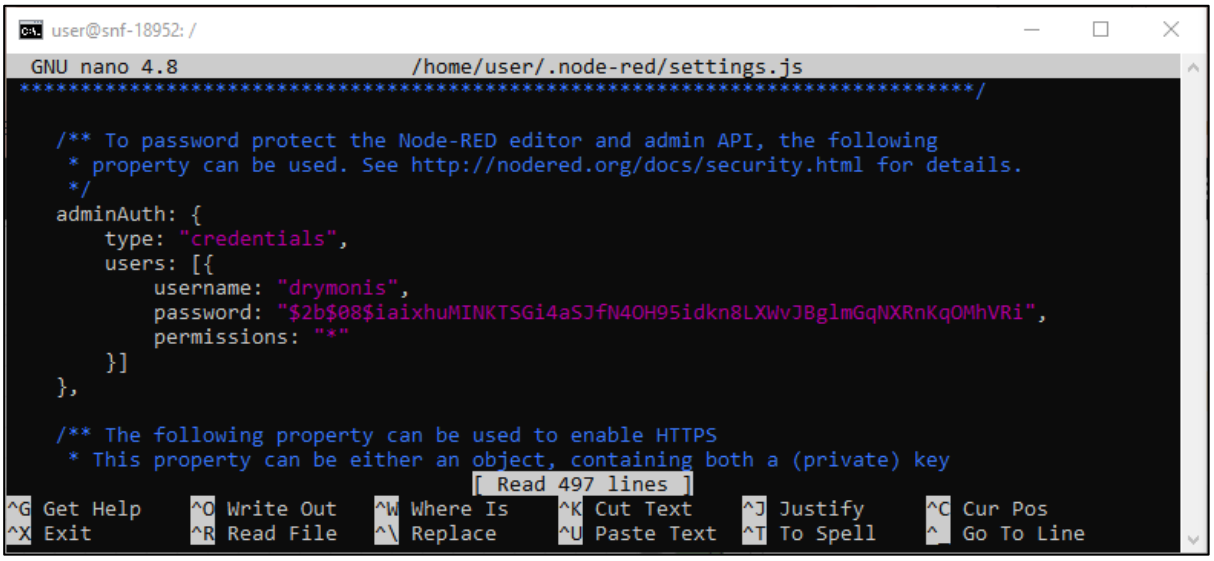


Εικόνα 3-76. Node-RED password hash.

Καταχωρήθηκε ο κωδικός χρήστη *msciot20004@* και αποδώθηκε το hash :

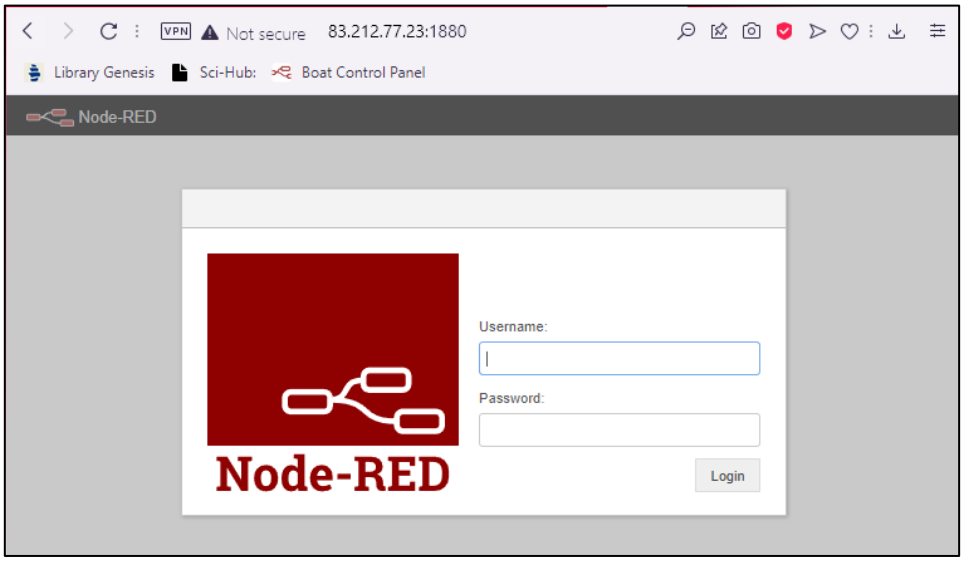
$2b$08$iaixhuMINKTSGi4aSJfN4OH95idkn8LXWvJBglmGqNXRnKqOMhVRi

5. Στη συνέχεια (Εικόνα 3-77), γίνεται αντιγραφή του hash κωδικού στο αρχείο settings.js στο οποίο έγινε επεξεργασία και στο βήμα 2 και 3. Ταυτόχρονα μπορεί να γίνει και αλλαγή του ονόματος χρήστη από *admin* σε *drymonis.*



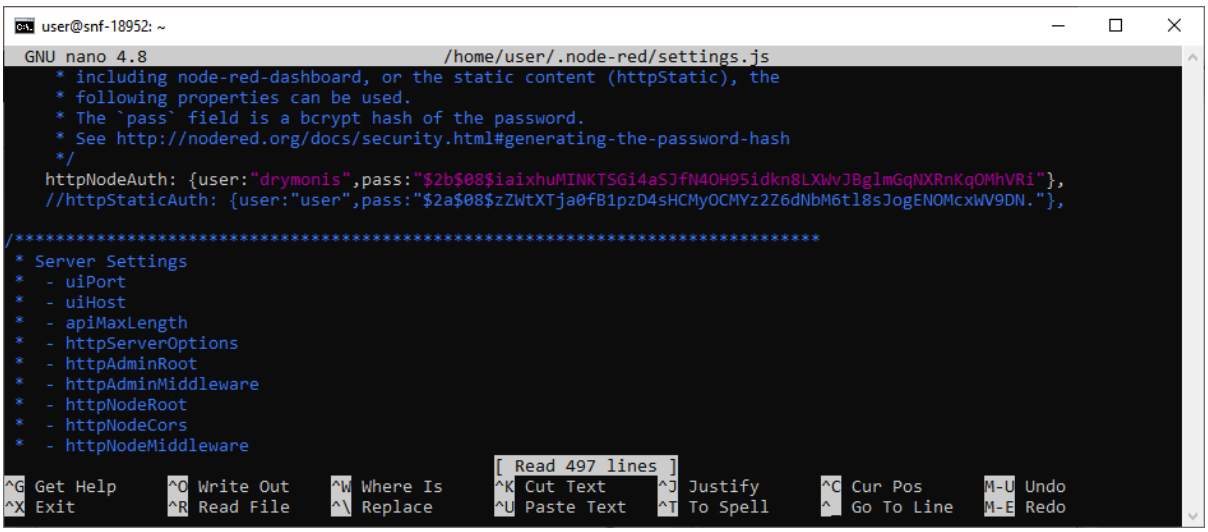
Εικόνα 3-77. Καταχώρηση του hash κωδικού στο αρχείο settings.js

6. Επιβεβαιώνεται ότι ζητούνται στοιχεία ταυτοποίησης κατά την πρόσβαση στην ιστοσελίδα του Node-RED (Εικόνα 3-78).



Εικόνα 3-78. Σελίδα πρόσβασης σε προγραμματιστικό περιβάλλον Node-RED.

7. Για την προσθήκη ταυτοποίησης του «Πίνακα ελέγχου του σκάφους» γίνεται παρόμοια διαδικασία επεξεργασίας του settings.js , αφαιρώντας τα σχόλια στις γραμμές που φαίνονται στην Εικόνα 3-79 και προσθέτοντας το ίδιο όνομα χρήστη και hashed κωδικό χρήστη με το προγραμματιστικό περιβάλλον του Node-Red.



Εικόνα 3-79. Επεξεργασία αρχείου settings.js.

8. Επιβεβαιώνεται ότι ζητούνται στοιχεία ταυτοποίησης και στον πίνακα ελέγχου (Εικόνα 3-80).

Εικόνα 3-80 Σελίδα πρόσβασης στον Πίνακα ελέγχου σκάφους

Στο «Παράρτημα Γ: Κώδικας ροών της πλατφόρμας Node-RED.» είναι διαθέσιμος όλος ο κώδικας (JSON) της εφαρμογής ελέγχου του μη επανδρωμένου σκάφους επιφανείας.

ΚΕΦΑΛΑΙΟ 4

Επίλογος

4.1 Εισαγωγή

Στο κεφάλαιο αυτό, γίνεται μια συνοπτική περιγραφή της υλοποίησης, η οποία περιλαμβάνει τις δοκιμές του σκάφους και τις παραμετροποιήσεις που έγιναν κατά την εκτέλεση τους. Επίσης, γίνεται αναφορά στα προβλήματα που προέκυψαν αλλά και οι διορθωτικές ενέργειες που έγιναν για την αντιμετώπισή τους. Επιπρόσθετα, αξιολογείται η συνολική απόδοση τόσο του σκάφους ως ολότητα, όσο και των επιμέρους υποσυστημάτων του. Καταλήγοντας, καταγράφονται προτάσεις βελτίωσης για τη διεύρυνση των μελλοντικών εφαρμογών, του συγκεκριμένου μη επανδρωμένου οχήματος.

4.2 Σύνοψη εργασίας

Στη συγκεκριμένη εργασία, σχεδιάστηκε και αναπτύχθηκε ένα ρομποτικό όχημα επιφανείας διαστάσεων 100x32x34 εκατοστά. Ως προς το κατασκευαστικό μέρος, παρουσιάστηκε αναλυτικά το μηχανολογικό κομμάτι, που περιλάμβανε τη συναρμολόγηση των συστημάτων πλοήγησης, τη δημιουργία βάσεων υποστήριξής των ηλεκτρονικών και ηλεκτρομηχανικών υποσυστημάτων και τη στεγανοποίηση του κύτους. Επίσης, παρουσιάστηκε η αρχιτεκτονική των ηλεκτρονικών υποσυστημάτων του οχήματος και έγινε η εγκατάσταση τους. Στο επόμενο στάδιο έγινε ο προγραμματισμός του αυτόματου πιλότου Ardupilot καθώς και του συστήματος επικοινωνιών που κάνει χρήση των τεχνολογιών IoT (MQTT, Node-RED PaaS, Cloud Computing, WiFi). Με την ολοκλήρωση του οχήματος, το σκάφος δοκιμάστηκε σε πραγματικές συνθήκες. Στις δοκιμές αυτές, το σκάφος είχε ως αποστολή την αυτόνομη εκτέλεση μίας σειράς διαδρομών καθώς και την επιστροφή του στην αρχική του θέση (Home ή Launch position), μέσω της εντολής ασφαλείας «RTL» (Return to Launch). Παράλληλα, δόθηκε βαρύτητα στη δοκιμή του γραφικού περιβάλλοντος ελέγχου, μέσω της πλατφόρμας υπηρεσιών Node-RED. Σκοπός των δοκιμών, ήταν να αξιολογηθεί η απόκριση και η αποτελεσματικότητα των τεχνολογιών του διαδικτύου των πραγμάτων που χρησιμοποιήθηκαν σε αυτή την υλοποίηση, για τον έλεγχο του μη επανδρωμένου οχήματος.

4.3 Προβλήματα-Αντιμετώπιση

Κατά τη διάρκεια των δοκιμών, διαπιστώθηκε ότι η εμβέλεια ελέγχου του οχήματος μέσω του WiFi (ESP32 και ESP8266), ήταν περιορισμένη στα 150 μέτρα περίπου. Αυτή η εμβέλεια, μπορούσε να μεταβληθεί αισθητά από τις καιρικές συνθήκες που επηρεάζουν άμεσα την επιφάνεια της θάλασσας. Επομένως, η επιφάνεια του νερού, αντανακλά τα ηλεκτρομαγνητικά κύματα και σχετίζεται άμεσα με το φαινόμενο της πολυοδικής διάθλασης σήματος (multipath fading). Για να βελτιωθεί η εμβέλεια σε αυτή τη μπάντα συχνοτήτων (2.4GHz), τοποθετήθηκε η συσκευή WiFi Access Point στο υψηλότερο σημείο της περιοχής δοκιμών [58]. Η ενέργεια αυτή αντιμετώπισε και ένα άλλο πρόβλημα που έχει να κάνει με τις ζώνες Fresnel κατά τη μετάδοση του σήματος μεταξύ δύο σημείων σε οπτική επαφή.

Το σκάφος κατά την αυτόνομη εκτέλεση διαδρομών, εμφάνιζε ταλαντώσεις στην κατεύθυνση της πορείας του. Για την αντιμετώπιση του προβλήματος αυτού, αρχικά εξετάστηκε εάν προέρχεται από μηχανολογικά αίτια. Επομένως, για να διαπιστωθεί η φύση του προβλήματος, το όχημα οδηγήθηκε από το χειριστή σε θάλασσα που δεν είχε κυματισμό. Εφόσον το όχημα δεν εμφάνιζε αστάθεια κατά τον έλεγχο του μέσω τηλεχειρισμού, δόθηκε έμφαση στη παραμετροποίηση των ελεγκτών PID για το σύστημα πρόωσης και του πηδαλίου από το λογισμικό Mission Planner όπως φαίνεται στην Εικόνα 3-59.

4.4 Παρατηρήσεις-Συμπεράσματα

Ο δείκτης “Communications Packet Loss” αφορά την ποιότητα επικοινωνίας των διαύλων UART, I2C, SPI και CAN του ελεγκτή πλοήγησης που ενσωματώνεται στο όχημα. Σε όλη τη διάρκεια των δοκιμών, ο δείκτης αυτός παρέμεινε μηδενικός, υποδεικνύοντας ότι οι συνδεσμολογίες μεταξύ των ηλεκτρονικών υποσυστημάτων ήταν άρτιες και χωρίς ηλεκτρομαγνητικές παρεμβολές.

Ο έλεγχος, του μη επανδρωμένου σκάφους επιφανείας, μέσω του γραφικού περιβάλλοντος που δημιουργήθηκε στην πλατφόρμα υπηρεσιών Node-RED, ήταν επιτυχής και λειτούργησαν σωστά όλες οι εντολές και οι ενδείξεις.

Η άμεση και ορθή ανταπόκριση του οχήματος σε κάθε εντολή, χωρίς να χρειαστεί να επαναληφθεί η αποστολή τους, δείχνει ότι το πρωτόκολλο MQTT είναι αξιόπιστο για τον έλεγχο μη επανδρωμένων οχημάτων.

Επίσης διαπιστώθηκε στην πράξη, ότι το Node-RED δίνει μεγάλη ευελιξία για τη δημιουργία σχετικών εφαρμογών ελέγχου και αυτοματοποίησης, καθώς έχει τη δυνατότητα να συνδυάσει πολλές διαφορετικές τεχνολογίες της πληροφορικής και του διαδικτύου.

4.5 Προτάσεις μελλοντικής εξέλιξης

Το σκάφος, αποτελεί ένα πρότυπο που σχεδιάστηκε για να αποδείξει ό,τι είναι εφικτός ο έλεγχος και η αυτόνομη λειτουργία του με χρήση τεχνολογιών IoT. Η λειτουργικότητα μιας τέτοιας πρότυπης υλοποίησης, μπορεί να ενισχυθεί μέσω υλοποίησης δράσεων όπως οι ακόλουθες:

- Επιλογή, εναλλακτικής και αποδοτικότερης τεχνολογίας ασύρματης μετάδοσης, καθώς το WiFi δεν ενδείκνυται για τέτοιου τύπου εφαρμογές. Εάν η εφαρμογή του οχήματος, απαιτεί χαμηλή κατανάλωση ενέργειας και μεγάλη εμβέλεια, τα δίκτυα LPWAN και συγκεκριμένα το NB-IoT αποτελεί μια ιδανική λύση που βασίζεται στα υπάρχοντα και ολοένα αναπτυσσόμενα δίκτυα κινητής τηλεφωνίας. Σε περίπτωση που υπάρχει η ανάγκη μεταφοράς μεγαλύτερου όγκου δεδομένων, μεταξύ οχήματος και σταθμού βάσης, μπορούν να χρησιμοποιηθούν μονάδες πομποδεκτών που κάνουν χρήση του δικτύου 4G [59].
- Η χρήση του Ardupilot και του πρωτόκολλου MAVLink δίνει την δυνατότητα αναβάθμισης του μη επανδρωμένου οχήματος με πιο ισχυρά υπολογιστικά συστήματα “Companion Computer”, ανοίγοντας το δρόμο για πιο ανεπτυγμένες δυνατότητες. Η μηχανική μάθηση [60] είναι μία από αυτές τις τεχνολογίες που μπορεί να ενσωματωθεί σε ένα “Companion Computer” για να δώσει στο όχημα μεγαλύτερο επίπεδο αυτονομίας και επεκτασιμότητας.
- Ο εντοπισμός και αποφυγή εμποδίων από ένα μη επανδρωμένο σκάφος επιφανείας, έχει πολύ μεγάλο βαθμό δυσκολίας, καθώς η μορφολογία της επιφάνειας της θάλασσας είναι δυναμική και αλλάζει συνεχώς σε συνάρτηση με το χρόνο. Επομένως,

η χρήση αισθητήρων μέτρησης απόστασης που κάνουν χρήση υπερήχων ή Laser δεν λειτουργούν αποδοτικά. Ισχυρές υπολογιστικές πλατφόρμες, όπως η σειρά Jetson της Nvidia μπορούν να δώσουν στο όχημα την ικανότητα οπτικής αναγνώρισης, που δεν περιορίζεται μόνο στον εντοπισμό εμποδίων αλλά και στην διεύρυνση των εφαρμογών του σκάφους. Πιο συγκεκριμένα, η οπτική αναγνώριση μπορεί να εφαρμοστεί στην κατηγοριοποίηση αντικειμένων ή έμβιων [61] οργανισμών.

- Βελτίωση της ενεργειακής αυτονομίας του οχήματος, ενσωματώνοντας ανανεώσιμές πηγές ενέργειας και ηλεκτρονικές διατάξεις διαχείρισης ισχύος. Τέτοιες πηγές, μπορεί να είναι φωτοβολταϊκά πάνελ [62], ανεμογεννήτριες [63], συστήματα εκμετάλλευσής της ενέργειας των υδάτινων κυμάτων και ρευμάτων [64] και η χρήση νάνο-υφασμάτων που εκμεταλλεύονται φαινόμενα όπως ο πιεζοηλεκτρισμός και ο τριβοηλεκτρισμός [65].
- Δημιουργία νέων κόμβων (Nodes) και εξειδικευμένων εργαλείων, για το περιβάλλον προγραμματισμού Node-RED, μπορεί να βοηθήσει στην ανάπτυξη βελτιωμένων και πιο εύχρηστων εφαρμογών, που αφορούν τον έλεγχο του μη επανδρωμένου σκάφους επιφανείας.
- Η χρήση προηγμένων αλγορίθμων ελέγχου μέσω εγκεφαλικών σημάτων μπορεί να καταστήσει δυνατό τον τηλε-έλεγχο του σκάφους ακόμα και από άτομα που έχουν αδυναμία χρήσης των άνω άκρων τους [66].
- Συνδυασμένη χρήση με ασύρματα δίκτυα αισθητήρων (Wireless Sensor Networks – WSNs) που αποτελούνται από πλωτούς ασύρματος κόμβους μπορεί να επαυξήσει τις δυνατότητες του σκάφους και το πλήθος των δυνητικών εφαρμογών του [67].

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

[1] "IEEE SPECTRUM FOR THE TECHNOLOGY INSIDER," [Online]. Available: https://spectrum.ieee.org/earthbound-robots-today-need-to-take-flight. [Accessed 2 2022].

[2] "IEEE SPECTRUM FOR THE TECHNOLOGY INSIDER," [Online]. Available: https://spectrum.ieee.org/ai-army-robots. [Accessed 2 2022].

[3] A. S. Gillis, "TechTarget," 3 2022. [Online]. Available: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT#. [Accessed 4 2022].

[4] D. Nilanjan, H. Aboul Ella, B. Chintan, A. Amira and S. Suresh Chandra, Internet of things and big data analytics toward next-generation intelligence, Cham, Switzerland: Springer, 2018.

[5] A. Baker, "Wilson Amplifiers," 1 2 2021. [Online]. Available: https://www.wilsonamplifiers.com/blog/the-difference-between-4g-lte-and-5g/. [Accessed 2 2022].

[6] K. L. Lueth, "IoT Analytics," 19 12 2014. [Online]. Available: https://iot-analytics.com/internet-of-things-definition/. [Accessed 2 2022].

[7] "IEEE Innovation at work," [Online]. Available: https://innovationatwork.ieee.org/autonomous-vehicles-for-today-and-for-the-future/. [Accessed 2 2022].

[8] "amazon," [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011. [Accessed 2 2022].

[9] M. Xie, FUNDAMENTALS OF ROBOTICS, Singapore: World Scientific Publishing Co. Pte. Ltd., 2003.

[10] B. John, BUILDING YOUR OWN DRONES A Begginer's Guide to Drones, UAV's and ROV's, Indianapolis: QUE, 2016.

[11] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/General_Atomics_MQ-9_Reaper#/media/File:MQ-9_Reaper_UAV_(cropped).jpg. [Accessed 2 2022].

[12] [Online]. Available: https://dvzpv6x5302g1.cloudfront.net/AcuCustom/Sitename/DAM/036/33073_original.jpg. [Accessed 2 2022].

[13] Clearath Robotics Inc., "AZO ROBOTICS," 26 3 2014. [Online]. Available: https://www.azorobotics.com/Article.aspx?ArticleID=177. [Accessed 2 2022].

[14] M. Justin E., Unmanned Surface Vehicles, 15 Years of Developement, Quebec: IEEE, OCEANS 2008, 2008.

[15] L. Zhixiang, Z. Youmin, Y. Xiang and Y. Chi, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control,* p. 25, 25 2 2016.

[16] M. Somayya, T. Siddharth and R. R, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications,* p. 10, 4 2015.

[17] K. ASHTON, "That 'Internet of Things' Thing," *RFID JOURNAL,* 22 6 2009.

[18] A. King, Programming the Internet of Things - An Introduction to Building Integrated, Device-to-Cloud IoT Solutions, Sebastopol, United States: O'Reilly Media, Inc, 2021.

[19] "Okeanos - Knossos," [Online]. Available: https://okeanos-knossos.grnet.gr/home/. [Accessed 2 2022].

[20] S.Satyanarayana, "CLOUD COMPUTING : SAAS," Computer Sciences and Telecommunications, Sri Venkateswara, 2012.

[21] W. Kim, "Cloud Computing: Today and Tomorrow," *JOURNAL OF OBJECT TECHNOLOGY,* p. Vol.8, 2 2009.

[22] M. J.Kavis, Architecting the Cloud - Design Decisions for Cloud Computing Service Models (SaaS, PaaS, IaaS), Canada: Wiley, 2014.

[23] Editors of Encyclopaedia, "Britannica," [Online]. Available: https://www.britannica.com/technology/Wi-Fi. [Accessed 2 2022].

[24] "WiFi Alliance," [Online]. Available: https://www.wi-fi.org/. [Accessed 2 2022].

[25] X. Liu, T. Zhang, N. Hu, P. Zhang and Y. Zhang, "The method of Internet of Things access and network communication based on MQTT," *Computer Communications,* 3 2020.

[26] "IPCisco," [Online]. Available: https://ipcisco.com/lesson/wireless-security-protocols/. [Accessed 2 2022].

[27] K. Monika, S. Vidushi and G. Neeti, "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things," 2018.

[28] G. Carlos A., M.-L. William and V. Marcelo, "Human-Robot Collaboration Based on Cyber-Physical Production System and MQTT," Procedia Manufacturing, 2020.

[29] "MAVLINK," [Online]. Available: http://mavlink.io/en/. [Accessed 2 2022].

[30] K. Anis, A. Azza, A. Maram Mohamed, J. Yasir, B. Abdelfettah and K. Mohamed, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," *IEEE Access,* p. 23, 2019.

[31] "BMFA Handbook," [Online]. Available: https://handbook.bmfa.uk/annex-b/7-rc-technical-info. [Accessed 2 2022].

[32] "International Telecommunication Union," 12 2018. [Online]. Available: https://www.itu.int/dms_pub/itu-r/md/15/wrs18/sp/R15-WRS18-SP-0003!!PDF-E.pdf. [Accessed 2 2022].

[33] T. Don, Principles of Spread-Spectrum Communication Systems, 4th ed., Springer, 2018.

[34] "Ardupilot," [Online]. Available: https://ardupilot.org/plane/docs/common-rc-systems.html. [Accessed 2 2022].

[35] "FrSKY," [Online]. Available: https://www.frsky-rc.com/frsky-advanced-communication-control-elevated-spread-spectrum-access-protocol-release/. [Accessed 3 2022].

[36] "HorusRC," [Online]. Available: https://www.horusrc.com/en/blog/accst-frsky/. [Accessed 2 2022].

[37] "Ardupilot," [Online]. Available: https://ardupilot.org/copter/docs/common-dshot-escs.html. [Accessed 3 2022].

[38] "Ardupilot," [Online]. Available: https://ardupilot.org/dev/docs/companion-computers.html. [Accessed 3 2022].

[39] E. Emad, S. Martin and H. T. Kristian, "A Survey of Open-Source UAV Flight Controllers and Flight Simulators," *Microprocessors and Microsystems,* 5 2018.

[40] "Node RED Programming Guide," [Online]. Available: http://noderedguide.com/. [Accessed 3 2022].

[41] M.-P. Alexandra, S.-R. David and A.-G. Itziar, "An Internet of Thing Architecture Based on Message Queuing Telemetry Transport Protocol and Node-RED: A Case Study for Monitoring Radon Gas," *Smart Cities,* p. 16, 5 2021.

[42] A. Gregoriou, "CYPRUS SAILING TV," 2 3 2019. [Online]. Available: https://cyprussailingtv.com/genikoi-naypigikoi-oroi-kai-technika-charaktiristika/. [Accessed 3 2022].

[43] C. Tim, "Marine-Pilots," 17 7 2020. [Online]. Available: https://www.marine-pilots.com/articles/84506-review-of-ships-pivot-point-science-maths-and-observation-where-is-centre-of-ships-rotation. [Accessed 3 2022].

[44] B. Vedder, "VESC," [Online]. Available: https://vesc-project.com/. [Accessed 3 2022].

[45] "Ardupilot," [Online]. Available: https://ardupilot.org/copter/docs/common-escs-and-motors.html. [Accessed 3 2022].

[46] "ST," [Online]. Available: https://www.st.com/en/applications/industrial-motor-control/3-phase-field-oriented-control-foc.html. [Accessed 3 2022].

[47] H. F., M. J., T. M., R. L.M. and B. W., "A Wireless Micro Inertial Measurement Unit (IMU)," [Online]. Available: http://ais.informatik.uni-freiburg.de/publications/papers/hoeflinger12i2mtc.pdf. [Accessed 3 2022].

[48] "Thales," [Online]. Available: https://www.thalesgroup.com/en/worldwide/aerospace/topaxyz-inertial-measurement-unit-imu/infographic. [Accessed 3 2022].

[49] A. D. King, 1998. [Online]. Available: https://www.imar-navigation.de/downloads/papers/inertial_navigation_introduction.pdf. [Accessed 3 2022].

[50] G.-E. Demoz, H. R.C. and P. J.D., "A Low Cost GPS/Inertial Attitude Heading Reference System (AHRS) for General Aviation Applications," *IEEE Xplore,* 5 1998.

[51] "GEM elettronica," [Online]. Available: https://www.gemrad.com/guidance-navigation-positioning/naval-navigation/#POLARISFOGFAMILY. [Accessed 3 2022].

[52] L. Wangyan, W. Zidong, W. Guoliang, M. Lifeng, H. Jun and D. Derui, "A Survey on Multisensor Fusion and Consensus Filtering for Sensor Networks," *Hindawi: Discrete Dynamics in Nature and Society,* 9 2015.

[53] N. Jens, Z. Anatolij and R. Jens-Peter, "The Impact of Adjacent Channel Interference in Multi-Radio Systems using IEEE 802.11," *IEEE Xplore,* 8 2008.

[54] "OpenTX," [Online]. Available: https://www.open-tx.org/. [Accessed 3 2022].

[55] F. Syed Abdullah, A. Sani Iyal, M. Mokhairi and J. Azrul Amri, "Robotic Indoor Path Planning Using Dijkstra's Algorithm with Multi-Layer Dictionaries," *IEEE Xplore,* 12 2015.

[56] olliw42, "github/olliw42/fastmavlink," [Online]. Available: https://github.com/olliw42/fastmavlink. [Accessed 3 2022].

[57] 256dpi, "github/256dpi/arduino-mqtt," [Online]. Available: https://github.com/256dpi/arduino-mqtt. [Accessed 3 2022].

[58] H. Arafat and M. Sangman, "Wireless Channel Models for Over-the-Sea Communication: A Comparative Study," *Applied Sciences,* p. 32, 17 1 2019.

[59] G. Vos, "What is Narrowband IoT (NB-IoT)," Sierra Wireless, 18 2 2022. [Online]. Available: https://www.sierrawireless.com/iot-blog/what-is-nb-iot/. [Accessed 15 5 2022].

[60] F. Zantalis, G. Koulouras, S. Karabetsos and D. Kandris, "A review of machine learning and IoT in smart transportation," Future Internet, 2019.

[61] Y. Huang, H. Wang, J. Ma, J. Lou and H. Yi, "Research and Practical Exploration of Test and Validation Technologies Applied on Unmanned Surface Vehicle Optical Recognition," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, 2021.

[62] C. Koukouvaos, D. Kandris and M. Samarakou, "Computer-Aided Modelling and Analysis of PV Systems: A Comparative Study," *The Scientific World Journal,* 2014.

[63] S. Theodoropoulos, D. Kandris, M. Samarakou and G. Koulouras, "Fuzzy Regulator Design for Wind Turbine Yaw Control," *The Scientific World Journal,* 2014.

[64] J. Duan, "A Multidimensional Wave Energy Fishing Vessel Energy Supply Device based on LMMHD Power Generation," *International Core Journal of Engineering,* pp. 168-175, 2022.

[65] D. Patricia I., "Energy Harvesting Materials and Structures for Smart Textile Applications: Recent Progress and Path Forward," *Sensors,* p. 27, 20 9 2021.

[66] N. Korovesis, D. Kandris , G. Koulouras and A. Alexandridis , "Robot Motion Control via an EEG-Based Brain–Computer Interface by Using Neural Networks and Alpha Brainwaves," *Electronics,* 2019.

[67] D. Kandris, C. Nakas , D. Vomvas and G. Koulouras, "Applications of Wireless Sensor Networks: An Up-to-Date Survey," *Applied System Innovation,* 2020.

Παράρτημα Α: Πίνακας ρυθμίσεων Ardupilot (μενού "Full Parameter List")

Command	Value	Units	Options	Description
ARMING_CHECK	30		0:None 1:All 2:Barometer 4:Compass 8:GPS Lock 16:INS(INertial Sensors - accels & gyros) 32:Parameters(unused) 64:RC Channels 128:Board voltage 256:Battery Level 1024:LoggingAvailable 2048:Hardware safety switch 4096:GPS configuration 8192:System 16384:Mission 32768:RangeFinder 65536:Camera 131072:AuxAuth 524288:FFT	Checks prior to arming motor. This is a bitmask of checks that will be performed before allowing arming. For most users it is recommended to leave this at the default of 1 (all checks enabled). You can select whatever checks you prefer by adding together the values of each check type to set this parameter. For example, to only allow arming when you have GPS lock and no RC failsafe you would set ARMING_CHECK to 72.
ARMING_REQUIRE	1		0:Disabled 1:THR_MIN PWM when disarmed 2:0 PWM when disarmed	Arming disabled until some requirements are met. If 0, there are no requirements (arm immediately). If 1, require rudder stick or GCS arming before arming motors and sends the minimum throttle PWM value to the throttle channel when disarmed. If 2, require rudder stick or GCS arming and send 0 PWM to throttle channel when disarmed. See the ARMING_CHECK_* parameters to see what checks are done before arming. Note, if setting this parameter to 0 a reboot is required to arm the plane. Also note, even with this parameter at 0, if ARMING_CHECK parameter is not also zero the plane may fail to arm throttle at boot due to a pre-arm check failure.
ARMING_RUDDER	2		0:Disabled 1:ArmingOnly 2:ArmOrDisarm	Allow arm/disarm by rudder input. When enabled arming can be done with right rudder, disarming with left rudder. Rudder arming only works in manual throttle modes with throttle at zero +- deadzone (RCx_DZ)
AVOID_ACCEL_MAX	3	m/s/s	0 9	Maximum acceleration with which obstacles will be avoided with. Set zero to disable acceleration limits
AVOID_BACKUP_DZ	0.1	m	0 2	Distance beyond AVOID_MARGIN parameter, after which vehicle will backaway from obstacles. Increase this parameter if you see vehicle going back and forth in front of obstacle.
AVOID_BACKUP_SPD	0.75	m/s	0 2	Maximum speed that will be used to back away from obstacles in GPS modes (m/s). Set zero to disable
AVOID_ENABLE	1		0:None 1:UseFence 2:UseProximitySensor 3:UseFence and UseProximitySensor 4:UseBeaconFence 7:All	Enabled/disable avoidance input sources
AVOID_MARGIN	2	m	1 10	Vehicle will attempt to stay at least this distance (in meters) from objects while in GPS modes
BATT_AMP_PERVLT	80	A/V		Number of amps that a 1V reading on the current sensor corresponds to. With a Pixhawk using the 3DR Power brick this should be set to 17. For the Pixhawk with the 3DR 4in1 ESC this should be 17.
BATT_ARM_MAH	0	mAh		Battery capacity remaining which is required to arm the aircraft. Set to 0 to allow arming at any capacity. Note that execept for smart batteries rebooting the vehicle will always reset the remaining capacity estimate, which can lead to this check not providing sufficent protection, it is

Command	Value	Units	Options	Description
				recommended to always use this in conjunction with the BATT__ARM_VOLT parameter.
BATT_ARM_VOLT	0	V		Battery voltage level which is required to arm the aircraft. Set to 0 to allow arming at any voltage.
BATT_BUS	0		0 3	Battery monitor I2C bus number
BATT_CAPACITY	5400	mAh		Capacity of the battery in mAh when full
BATT_CRT_MAH	0	mAh		Battery capacity at which the critical battery failsafe is triggered. Set to 0 to disable battery remaining failsafe. If the battery capacity drops below this level the vehicle will perform the failsafe specified by the BATT__FS_CRT_ACT parameter.
BATT_VOLT_MULT	11			Used to convert the voltage of the voltage sensing pin (BATT_VOLT_PIN) to the actual battery's voltage (pin_voltage * VOLT_MULT). For the 3DR Power brick with a Pixhawk, this should be set to 10.1. For the Pixhawk with the 3DR 4in1 ESC this should be 12.02. For the PX using the PX4IO power supply this should be set to 1.
BRD_SAFETYOPTION	0			This controls the activation of the safety button. It allows you to control if the safety button can be used for safety enable and/or disable, and whether the button is only active when disarmed
EK3_GPS_CHECK	31			This is a 1 byte bitmap controlling which GPS preflight checks are performed. Set to 0 to bypass all checks. Set to 255 perform all checks. Set to 3 to check just the number of satellites and HDoP. Set to 31 for the most rigorous checks that will still allow checks to pass when the copter is moving, eg launch from a boat.
FENCE_ACTION	1		0:Report Only 1:RTL or Hold 2:Hold 3:SmartRTL or RTL or Hold 4:SmartRTL or Hold	What action should be taken when fence is breached
FENCE_ENABLE	1		0:Disabled 1:Enabled	Allows you to enable (1) or disable (0) the fence functionality
FENCE_MARGIN	2	m	1 10	Distance that autopilot's should maintain from the fence to avoid a breach
FENCE_RADIUS	300	m	30 10000	Circle fence radius which when breached will cause an RTL
FENCE_TYPE	6		0:None 1:Altitude 2:Circle 3:Altitude and Circle 4:Polygon 5:Altitude and Polygon 6:Circle and Polygon 7:All	Enabled fence types held as bitmask
FRAME_CLASS	2		0:Undefined 1:Rover 2:Boat 3:BalanceBot	Frame Class
GPS_GNSS_MODE	77		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51: GPS+IMES+QZSS+SBAS (Japan Only) 64:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the first GPS (all unchecked or zero to leave GPS as configured)
GPS_GNSS_MODE2	77		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51: GPS+IMES+QZSS+SBAS (Japan Only) 64:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the second GPS (all unchecked or zero to leave GPS as configured)
GPS_RATE_MS	200	ms	50 200100:10Hz 125:8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz(default) is not allowed. Raising the rate above 5Hz usually provides little benefit and for some GPS (eg Ublox M9N) can severely impact performance.

Command	Value	Units	Options	Description
GPS_RATE_MS2	200	ms	50 200100:10Hz 125:8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz(default) is not allowed. Raising the rate above 5Hz usually provides little benefit and for some GPS (eg Ublox M9N) can severely impact performance.
INITIAL_MODE	0		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	This selects the mode to start in on boot. This is useful for when you want to start in AUTO mode on boot without a receiver. Usually used in combination with when AUTO_TRIGGER_PIN or AUTO_KICKSTART.
MODE_CH	6			RC Channel to use for driving mode control
MODE1	0		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 1 (910 to 1230 and above 2049)
MODE2	0		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 2 (1231 to 1360)
MODE3	10		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 3 (1361 to 1490)
MODE4	10		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 4 (1491 to 1620)
MODE5	11		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 5 (1621 to 1749)
MODE6	11		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Driving mode for switch position 6 (1750 to 2049)
OA_TYPE	2		0:Disabled 1:BendyRuler 2:Dijkstra 3:Dijkstra with BendyRuler	Enabled/disable path planning around obstacles
RC5_OPTION	41		0:Do Nothing 4:RTL 5:Save Trim (4.1 and lower) 7:Save WP 9:Camera Trigger 11:Fence 16:Auto 19:Gripper 24:Auto Mission Reset 27:Retract Mount 28:Relay On/Off 30:Lost Rover Sound 31:Motor Emergency Stop 34:Relay2 On/Off 35:Relay3 On/Off 36:Relay4 On/Off 40:Proximity Avoidance 41:ArmDisarm (4.1 and lower) 42:SmartRTL 46:RC Override Enable 50:LearnCruise 51:Manual 52:Acro 53:Steering 54:Hold 55:Guided 56:Loiter 57:Follow 58:Clear Waypoints 59:Simple Mode 62:Compass Learn 63:Sailboat Tack 65:GPS Disable 66:Relay5 On/Off 67:Relay6 On/Off 74:Sailboat motoring 3pos 78:RunCam Control 79:RunCam OSD Control 80:Viso Align 81:Disarm 90:EKF Pos Source 94:VTX Power 97:Windvane home heading direction offset 100:KillIMU1 101:KillIMU2 102:Camera Mode Toggle 105:GPS Disable Yaw 106:Disable Airspeed Use 153:ArmDisarm (4.2 and higher) 155:set steering trim to current servo and RC 156:Torqeedo Clear Err 201:Roll 202:Pitch 207:MainSail 208:Flap 211:Walking Height 300:Scripting1 301:Scripting2 302:Scripting3 303:Scripting4 304:Scripting5 305:Scripting6 306:Scripting7 307:Scripting8	Function assigned to this RC channel

Command	Value	Units	Options	Description
SERIAL1_BAUD	57		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 230:230400 256:256000 460:460800 500:500000 921:921600 1500:1500000	The baud rate used on the Telem1 port. Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL1_OPTIONS	7			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire. The Swap option allows the RX and TX pins to be swapped on STM32F7 based boards.
SERIAL1_PROTOCOL	10		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Rangefinder 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 13:Beacon 14:Volz servo out 15:SBus servo out 16:ESC Telemetry 17:Devo Telemetry 18:OpticalFlow 19:RobotisServo 20:NMEA Output 21:WindVane 22:SLCAN 23:RCIN 24:MegaSquirt EFI 25:LTM 26:RunCam 27:HottTelem 28:Scripting 29:Crossfire VTX 30:Generator 31:Winch 32:MSP 33:DJI FPV 34:AirSpeed 35:ADSB 36:AHRS 37:SmartAudio 38:FETtecOneWire 39:Torqeedo 40:AIS 41:CoDevESC 42:DisplayPort 43:MAVLink High Latency	Control what protocol to use on the Telem1 port. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL2_BAUD	921		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 230:230400 256:256000 460:460800 500:500000 921:921600 1500:1500000	The baud rate of the Telem2 port. Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL2_OPTIONS	0			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire.
SERIAL2_PROTOCOL	2		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Rangefinder 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 13:Beacon 14:Volz servo out 15:SBus servo out 16:ESC Telemetry 17:Devo Telemetry 18:OpticalFlow 19:RobotisServo 20:NMEA Output 21:WindVane 22:SLCAN 23:RCIN 24:MegaSquirt EFI 25:LTM 26:RunCam 27:HottTelem 28:Scripting 29:Crossfire VTX 30:Generator 31:Winch 32:MSP 33:DJI FPV 34:AirSpeed 35:ADSB 36:AHRS 37:SmartAudio 38:FETtecOneWire 39:Torqeedo 40:AIS 41:CoDevESC 42:DisplayPort 43:MAVLink High Latency	Control what protocol to use on the Telem2 port. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL3_BAUD	38		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 230:230400 256:256000	The baud rate used for the Serial 3 (GPS). Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and

Command	Value	Units	Options	Description
			460:460800 500:500000 921:921600 1500:1500000	then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL3_OPTIONS	0			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire.
SERIAL3_PROTOCOL	5		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Rangefinder 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 13:Beacon 14:Volz servo out 15:SBus servo out 16:ESC Telemetry 17:Devo Telemetry 18:OpticalFlow 19:RobotisServo 20:NMEA Output 21:WindVane 22:SLCAN 23:RCIN 24:MegaSquirt EFI 25:LTM 26:RunCam 27:HottTelem 28:Scripting 29:Crossfire VTX 30:Generator 31:Winch 32:MSP 33:DJI FPV 34:AirSpeed 35:ADSB 36:AHRS 37:SmartAudio 38:FETtecOneWire 39:Torqeedo 40:AIS 41:CoDevESC 42:DisplayPort 43:MAVLink High Latency	Control what protocol Serial 3 (GPS) should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL5_BAUD	57		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 230:230400 256:256000 460:460800 500:500000 921:921600 1500:1500000	The baud rate used for Serial5. Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL5_OPTIONS	0			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire.
SERIAL5_PROTOCOL	1		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Rangefinder 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 13:Beacon 14:Volz servo out 15:SBus servo out 16:ESC Telemetry 17:Devo Telemetry 18:OpticalFlow 19:RobotisServo 20:NMEA Output 21:WindVane 22:SLCAN 23:RCIN 24:MegaSquirt EFI 25:LTM 26:RunCam 27:HottTelem 28:Scripting 29:Crossfire VTX 30:Generator 31:Winch 32:MSP 33:DJI FPV 34:AirSpeed 35:ADSB 36:AHRS 37:SmartAudio 38:FETtecOneWire 39:Torqeedo 40:AIS 41:CoDevESC 42:DisplayPort 43:MAVLink High Latency	Control what protocol Serial5 port should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL7_BAUD	57		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 230:230400 256:256000 460:460800 500:500000 921:921600 1500:1500000	The baud rate used for Serial7. Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL7_OPTIONS	0			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX

Command	Value	Units	Options	Description
				option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire.
SERIAL7_PROTOCOL	-1		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Rangefinder 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 13:Beacon 14:Volz servo out 15:SBus servo out 16:ESC Telemetry 17:Devo Telemetry 18:OpticalFlow 19:RobotisServo 20:NMEA Output 21:WindVane 22:SLCAN 23:RCIN 24:MegaSquirt EFI 25:LTM 26:RunCam 27:HottTelem 28:Scripting 29:Crossfire VTX 30:Generator 31:Winch 32:MSP 33:DJI FPV 34:AirSpeed 35:ADSB 36:AHRS 37:SmartAudio 38:FETtecOneWire 39:Torqeedo 40:AIS 41:CoDevESC 42:DisplayPort 43:MAVLink High Latency	Control what protocol Serial7 port should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.
SERVO1_FUNCTION	26		-1:GPIO 0:Disabled 1:RCPassThru 6:MountPan 7:MountTilt 8:MountRoll 9:MountOpen 10:CameraTrigger 12:Mount2Pan 13:Mount2Tilt 14:Mount2Roll 15:Mount2Open 22:SprayerPump 23:SprayerSpinner 26:GroundSteering 28:Gripper 33:Motor1 34:Motor2 35:Motor3 36:Motor4 51:RCIN1 52:RCIN2 53:RCIN3 54:RCIN4 55:RCIN5 56:RCIN6 57:RCIN7 58:RCIN8 59:RCIN9 60:RCIN10 61:RCIN11 62:RCIN12 63:RCIN13 64:RCIN14 65:RCIN15 66:RCIN16 70:Throttle 73:ThrottleLeft 74:ThrottleRight 88:Winch 89:Main Sail 90:CameraISO 91:CameraAperture 92:CameraFocus 93:CameraShutterSpeed 94:Script1 95:Script2 96:Script3 97:Script4 98:Script5 99:Script6 100:Script7 101:Script8 102:Script9 103:Script10 104:Script11 105:Script12 106:Script13 107:Script14 108:Script15 109:Script16 120:NeoPixel1 121:NeoPixel2 122:NeoPixel3 123:NeoPixel4 128:WingSailElevator 129:ProfiLED1 130:ProfiLED2 131:ProfiLED3 132:ProfiLEDClock 133:Winch Clutch 134:SERVOn_MIN 135:SERVOn_TRIM 136:SERVOn_MAX 137:SailMastRotation 138:Alarm 139:Alarm Inverted	Function assigned to this servo. Setting this to Disabled(0) will setup this output for control by auto missions or MAVLink servo set commands. any other value will enable the corresponding function
SERVO1_MAX	1900	PWM	800 2200	maximum PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
SERVO1_MIN	1100	PWM	500 2200	minimum PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
SERVO1_REVERSED	0		0:Normal 1:Reversed	Reverse servo operation. Set to 0 for normal operation. Set to 1 to reverse this output channel.
SERVO1_TRIM	1500	PWM	800 2200	Trim PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.

Command	Value	Units	Options	Description
SERVO3_FUNCTION	70		-1:GPIO 0:Disabled 1:RCPassThru 6:MountPan 7:MountTilt 8:MountRoll 9:MountOpen 10:CameraTrigger 12:Mount2Pan 13:Mount2Tilt 14:Mount2Roll 15:Mount2Open 22:SprayerPump 23:SprayerSpinner 26:GroundSteering 28:Gripper 33:Motor1 34:Motor2 35:Motor3 36:Motor4 51:RCIN1 52:RCIN2 53:RCIN3 54:RCIN4 55:RCIN5 56:RCIN6 57:RCIN7 58:RCIN8 59:RCIN9 60:RCIN10 61:RCIN11 62:RCIN12 63:RCIN13 64:RCIN14 65:RCIN15 66:RCIN16 70:Throttle 73:ThrottleLeft 74:ThrottleRight 88:Winch 89:Main Sail 90:CameraISO 91:CameraAperture 92:CameraFocus 93:CameraShutterSpeed 94:Script1 95:Script2 96:Script3 97:Script4 98:Script5 99:Script6 100:Script7 101:Script8 102:Script9 103:Script10 104:Script11 105:Script12 106:Script13 107:Script14 108:Script15 109:Script16 120:NeoPixel1 121:NeoPixel2 122:NeoPixel3 123:NeoPixel4 128:WingSailElevator 129:ProfiLED1 130:ProfiLED2 131:ProfiLED3 132:ProfiLEDClock 133:Winch Clutch 134:SERVOn_MIN 135:SERVOn_TRIM 136:SERVOn_MAX 137:SailMastRotation 138:Alarm 139:Alarm Inverted	Function assigned to this servo. Setting this to Disabled(0) will setup this output for control by auto missions or MAVLink servo set commands. any other value will enable the corresponding function
SERVO3_MAX	1900	PWM	800 2200	maximum PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
SERVO3_MIN	1100	PWM	500 2200	minimum PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
SERVO3_REVERSED	1		0:Normal 1:Reversed	Reverse servo operation. Set to 0 for normal operation. Set to 1 to reverse this output channel.
SERVO3_TRIM	1482	PWM	800 2200	Trim PWM pulse width in microseconds. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.

Παράρτημα Β: Κώδικας Μικροελεγκτή ESP32

```
1.  //Author: Theodor Drymonis
2.  //This code was created by combining the “fastmavlink” and “arduino-mqtt” libraries
3.  //-----------------------------
4.  // The arduino-mqtt library
5.  // https://github.com/256dpi/arduino-mqtt
6.  //-----------------------------
7.  // Based on arduino-mqtt Example:
8.  // ESP32DevelopmentBoardSecure
9.  //-----------------------------
10. // The fastMavlink library
11. // (c) OlliW, OlliW42, www.olliw.eu
12. // https://github.com/olliw42/fastmavlink/
13. //-----------------------------
14. // Based on fastMavlink Example:
15. // One Link - One Component
16. // The better2 implemenation.
17. // For details see fastMavlink github repo.
18. // For Arduino IDE.
19. //-----------------------------
20. // Licence: This code is free and open and you can use it
21. // in whatever way you want. It is provided as is with no
22. // implied or expressed warranty of any kind.
23. //-----------------------------
24. // Do this before you start:
25. // Copy the fastMavlink c_library folder into the sketch folder.
26. //-----------------------------


27. //-----------------------------
28. // Implement the interface assumed by the fastMavlink examples
29. //-----------------------------
30. // Attention: The Arduino's buffer for serial write can be very limited (the docs say 64 bytes!),
31. // and can be too small for really doing MAVLink. The serial_has_space() function may hence trigger
32. // often and effectively block the code from working. In that case you may work around by letting
    it
33. // return true always


34. // Depending on board and core and setup, the main serial might be Serial, Serial1, ...
35. // Choose what's correct for you.


36. #define enable_debug_serial true


37. //###################### START//HARDWARE SERIAL PORT SETUP
38. #include <HardwareSerial.h>
```

```
39. HardwareSerial Mavlink_Serial(1);
40. #define RXD2 22
41. #define TXD2 5

42. //#define SERIAL  Serial1 //if the esp board support an extra serial port
43. #define SERIAL_BAUD 57600 // 57600 or 115200 are usually good choices
44. //###################### END//HARDWARE SERIAL PORT SETUP

45. // for string manipulation
46. #include <stdio.h>
47. #include <string.h>

48. //########################################## START//mavlink function global variables
49. uint8_t request_data_stream_once = 1; // enable or disable autopilot telemetry readings
50. uint8_t initiate_mission_once = 0;
51. int32_t mission_cmd_latitude = 383411471;  // safe initial coordinates
52. int32_t mission_cmd_longitude = 236798823; // safe initial coordinates
53. float mission_cmd_timestamp;
54. uint8_t arm_disarm_once = 1;
55. float arm_disarm_switch = 0; // 0 = disarm , 1 = arm
56. uint8_t rtl_once = 0;
57. uint8_t request_rtl_coordinates_once = 1;
58. uint8_t set_home_gps_coordinates_once = 0;
59. int32_t rtl_cmd_latitude = 383404429;  // safe initial coordinates
60. int32_t rtl_cmd_longitude = 236766368; // safe initial coordinates
61. float rtl_cmd_timestamp;
62. //########################################## END//mavlink function global variables

63. //########################################################## START//MQTT + WIFI
64. #include <WiFi.h>
65. #include <MQTT.h>

66. String gps_raw_data;
67. String mission_ack_data;
68. String command_ack_data;
69. String gps_rtl_data;
70. String mission_gps_data;

71. const char ssid[] = "GINA";
72. const char pass[] = "6974253002";

73. WiFiClient net;
74. MQTTClient client;

75. unsigned long lastMillis = 0;
```

```
76. void connect()
77. {
78. #if (enable_debug_serial == true)
79. Serial.print("checking wifi...");
80. #endif
81. while (WiFi.status() != WL_CONNECTED)
82. {
83. #if (enable_debug_serial == true)
84. Serial.print(".");
85. #endif
86. delay(1000);
87. }

88. #if (enable_debug_serial == true)
89. Serial.print("\n MQTT connecting...");
90. #endif
91. while (!client.connect("esp32_mavlink_mqtt_bridge", "drymonis", "msciot20004@"))
92. { //("mqtt_client_id","username","password")
93. #if (enable_debug_serial == true)
94. Serial.print(".");
95. #endif
96. delay(1000);
97. }

98. #if (enable_debug_serial == true)
99. Serial.println("\nconnected!");
100.#endif

101.client.subscribe("autopilot/mission/command/execute", 2);                 //("topic", qos)
102.client.subscribe("autopilot/mission/command/arm_disarm", 2);              //("topic", qos)
103.client.subscribe("autopilot/mission/command/waypoint", 2);                //("topic", qos)
104.client.subscribe("autopilot/mission/command/return_to_launch", 2);        //("topic", qos)
105.client.subscribe("autopilot/mission/command/request_rtl_coordinates", 2); //("topic", qos)
106.client.subscribe("autopilot/mission/command/set_rtl_coordinates", 2);     //("topic", qos)
107.}

108.void messageReceived(String &topic, String &payload)
109.{
110.#if (enable_debug_serial == true)
111.Serial.println("");
112.Serial.println("incoming MQTT message: " + topic + " - " + payload);
113.#endif

114.// Note: Do not use the client in the callback to publish, subscribe or
115.// unsubscribe as it may cause deadlocks when other things arrive while
```

```
// sending and receiving acknowledgments. Instead, change a global variable,
// or push to a queue and handle it in the loop after calling `client.loop()`.

if (topic == "autopilot/mission/command/arm_disarm")
{
#if (enable_debug_serial == true)
Serial.print("\nNew mqtt arm_disarm message: ");
#endif
if (payload == "0")
{
#if (enable_debug_serial == true)
Serial.print("switching to disarmed");
#endif
arm_disarm_switch = 0; // switch position to off
arm_disarm_once = 1;   // enable execute once flag
}
else if (payload == "1")
{
#if (enable_debug_serial == true)
Serial.print("switching to armed");
#endif
arm_disarm_switch = 1; // switch position to on
arm_disarm_once = 1;   // enable execute once flag
}
}

if (topic == "autopilot/mission/command/waypoint")
{

#if (enable_debug_serial == true)
Serial.print("\nNew mqtt waypoint message incomming...");
#endif
// start of //decode_gps_coordinates(float latitude, float logitude)
uint8_t array_length = payload.length(); // gives the String length minus the terminating zero
    contained in it (NULL). maximum lenth 255
array_length = array_length + 1;         // adding +1 to the charracter array length for the
    terminating zero space. This is required for proper string conversion
char payload_char_array[array_length];
payload.toCharArray(payload_char_array, array_length);

char *strings[4]; // an array of pointers to the pieces of the topic's mqtt payload after strtok()
    // the strings correspond to the number of variables expected to be included in message + a null
    string. Here we have(3+1)
char *ptr = NULL;

byte index = 0;                        // number of variables included in message
```

```
153.ptr = strtok(payload_char_array, ","); // delimiter
154.while (ptr != NULL)
155.{
156.strings[index] = ptr;
157.index++;
158.ptr = strtok(NULL, ",");
159.}
160.#if (enable_debug_serial == true)
161.// Serial.println(index);
162.//  print all the parts
163.Serial.println("The Pieces separated by strtok()");
164.for (int n = 0; n < index; n++)
165.{
166.Serial.print(n);
167.Serial.print("  ");
168.Serial.println(strings[n]);
169.}
170.#endif

171.// Those values are Latitude,Longitude,Timestamp.
172.//  use the atoi() and atof() functions to convert ASCII strings to numbers.
173.mission_cmd_latitude = atoi(strings[0]); //
    http://www.cplusplus.com/reference/cstdlib/atoi/?kw=atoi
174.mission_cmd_longitude = atoi(strings[1]);
175.mission_cmd_timestamp = atoi(strings[2]);
176.// end of //decode_gps_coordinates(float latitude, float logitude)

177.#if (enable_debug_serial == true)
178.Serial.print("\nNew mqtt mission coordinates message received !");
179.Serial.println();
180.Serial.print("\nLatitude = ");
181.Serial.println(mission_cmd_latitude);
182.Serial.print("Longitude = ");
183.Serial.println(mission_cmd_longitude);
184.Serial.print("Timestamp = ");
185.Serial.println(mission_cmd_timestamp);
186.#endif

187.initiate_mission_once = 1;
188.}

189.if (topic == "autopilot/mission/command/return_to_launch")
190.{
191.#if (enable_debug_serial == true)
192.Serial.print("\nNew mqtt return_to_launch message received !");
193.#endif
194.if (payload == "1")
```

```
{
rtl_once = 1;
}
}

if (topic == "autopilot/mission/command/set_rtl_coordinates")
{

#if (enable_debug_serial == true)
Serial.print("\nNew mqtt set_rtl_coordinates message incomming...");
#endif
// start of //decode_gps_coordinates(float latitude, float logitude)
uint8_t array_length = payload.length(); // gives the String length minus the terminating zero contained in it (NULL). maximum lenth 255
array_length = array_length + 1;         // adding +1 to the charracter array length for the terminating zero space. This is required for proper string conversion
char payload_char_array[array_length];
payload.toCharArray(payload_char_array, array_length);

char *strings[4]; // an array of pointers to the pieces of the topic's mqtt payload after strtok()
// the strings correspond to the number of variables expected to be included in message + a null string. Here we have(3+1)
char *ptr = NULL;

byte index = 0;                         // number of variables included in message
ptr = strtok(payload_char_array, ","); // delimiter
while (ptr != NULL)
{
strings[index] = ptr;
index++;
ptr = strtok(NULL, ",");
}
#if (enable_debug_serial == true)
// Serial.println(index);
//  print all the parts
Serial.println("The Pieces separated by strtok()");
for (int n = 0; n < index; n++)
{
Serial.print(n);
Serial.print("  ");
Serial.println(strings[n]);
}
#endif

// Those values are Latitude,Longitude,Timestamp.
//  use the atoi() and atof() functions to convert ASCII strings to numbers.
```

```
232.rtl_cmd_latitude = atoi(strings[0]); // http://www.cplusplus.com/reference/cstdlib/atoi/?kw=atoi
233.rtl_cmd_longitude = atoi(strings[1]);
234.rtl_cmd_timestamp = atoi(strings[2]);
235.// end of //decode_gps_coordinates(float latitude, float logitude)

236.#if (enable_debug_serial == true)
237.Serial.print("\nNew mqtt rtl coordinates message received !");
238.Serial.println();
239.Serial.print("\nLatitude = ");
240.Serial.println(rtl_cmd_latitude);
241.Serial.print("Longitude = ");
242.Serial.println(rtl_cmd_longitude);
243.Serial.print("Timestamp = ");
244.Serial.println(rtl_cmd_timestamp);
245.#endif

246.set_home_gps_coordinates_once = 1;
247.}

248.if (topic == "autopilot/mission/command/request_rtl_coordinates")
249.{
250.#if (enable_debug_serial == true)
251.Serial.print("\nNew mqtt request_rtl_coordinates message received !");
252.#endif
253.if (payload == "1")
254.{
255.request_rtl_coordinates_once = 1;
256.}
257.}
258.}
259.//########################################################## END//MQTT + WIFI

260.//########################################################## START//MAVLINK SERIAL FUNCTIONS
261.uint16_t serial_available(void)
262.{
263.uint16_t available = Mavlink_Serial.available();
264.// uint16_t available = SERIAL.available();
265.return (available > 0) ? available : 0;
266.}

267.void serial_read_char(uint8_t *c)
268.{
269.*c = Mavlink_Serial.read();
270.//*c = SERIAL.read();
271.}
```

```
uint8_t serial_has_space(uint16_t count)
{
return (Mavlink_Serial.availableForWrite() >= count) ? 1 : 0;
// return (SERIAL.availableForWrite() >= count) ? 1 : 0;
}

void serial_write_buf(uint8_t *buf, uint16_t len)
{
for (uint16_t i = 0; i < len; i++)
Mavlink_Serial.write(buf[i]);
// for(uint16_t i = 0; i < len; i++) SERIAL.write(buf[i]);
}

uint32_t get_time_ms()
{
return millis();
}

// let's do this in addition, have a LED on a pin to visually see some action
uint8_t BLINK_PIN = 5;
uint8_t blink = 0;

#define LED_TOGGLE                                  \
{                                                  \
digitalWrite(BLINK_PIN, (blink) ? HIGH : LOW); \
blink = (blink) ? 0 : 1;                        \
}

//------------------------------
// Include the fastMavlink library C code
//------------------------------
// For this example, common.xml is a good choice for the dialect,
// but choose whichever dialect you prefer.

#define FASTMAVLINK_SERIAL_WRITE_CHAR
void fmav_serial_write_char(uint8_t c)
{
Mavlink_Serial.write(c);
// SERIAL.write(c);
}
//########################################################## END//MAVLINK SERIAL FUNCTIONS

// The "..../common/common.h" is the way how fastMavlink wants it to be.
// If you would do ".../common/mavlink.h" like you would for the pymavlink-mavgen
// library, you would enable fastMavlink's pymavlink-mavgen mimicry. The code would
```

```
// still work, but we want to do fastMavlink here :).
#include "c_library/common/common.h"

// Arduino IDE does some automatic changes to the code. This prevents it inserting
// a function prototype in the wrong place.
void handleMessage(fmav_message_t *msg);

//------------------------------
// Main Code start
//------------------------------

uint8_t my_sysid = 1; // match it to the sys id of your autopilot

uint8_t my_compid = MAV_COMP_ID_PERIPHERAL; // match it to what your component wants to be

// this is needed to keep various info
fmav_status_t status;

// this holds the received message, and is also the working buffer for the parser
fmav_message_t msg;

// some variables we need

// uint8_t send_statustext = 0;

uint32_t tlast_ms = 0;

//################################################################################ START//request data stream command
//########################################## mavlink_msg_request_data_stream.h
// /*
// send the "Hello World" STATUSTEXT message
// returns 1 if successful, 0 else
uint8_t sendRequestDataStream(uint8_t request_data_stream_switch)
{
// We use here the payload structure and the xxx_encode_to_serial() function.
// This is for demonstrational purposes, but in case of STATUSTEXT it also makes sense
// as we need some buffer to deal with the text array anyways.

// create payload variable
fmav_request_data_stream_t payload; // you could name the payload object inside here whatever you like
```

```
// Request Data from Ardupilot

payload.req_message_rate = 0x01;  // number of times per second to request the data in hex
payload.target_system = my_sysid; //
payload.target_component = 0;     //
payload.req_stream_id = MAV_DATA_STREAM_ALL;
payload.start_stop = request_data_stream_switch; // 1 = start streaming, 0 = stop streaming

// STREAMS that can be requested
/*
        •   Definitions are in common.h: enum MAV_DATA_STREAM and more importantly at:
https://mavlink.io/en/messages/common.html#MAV_DATA_STREAM
*
        •   MAV_DATA_STREAM_ALL=0, // Enable all data streams
        •   MAV_DATA_STREAM_RAW_SENSORS=1, /* Enable IMU_RAW, GPS_RAW, GPS_STATUS packets.
        •   MAV_DATA_STREAM_EXTENDED_STATUS=2, /* Enable GPS_STATUS, CONTROL_STATUS, AUX_STATUS
        •   MAV_DATA_STREAM_RC_CHANNELS=3, /* Enable RC_CHANNELS_SCALED, RC_CHANNELS_RAW,
            SERVO_OUTPUT_RAW
        •   MAV_DATA_STREAM_RAW_CONTROLLER=4, /* Enable ATTITUDE_CONTROLLER_OUTPUT,
            POSITION_CONTROLLER_OUTPUT, NAV_CONTROLLER_OUTPUT.
        •   MAV_DATA_STREAM_POSITION=6, /* Enable LOCAL_POSITION, GLOBAL_POSITION/GLOBAL_POSITION_INT
            messages.
        •   MAV_DATA_STREAM_EXTRA1=10, /* Dependent on the autopilot
        •   MAV_DATA_STREAM_EXTRA2=11, /* Dependent on the autopilot
        •   MAV_DATA_STREAM_EXTRA3=12, /* Dependent on the autopilot
        •   MAV_DATA_STREAM_ENUM_END=13,
*
        •   Data in PixHawk available in:
        •   - Battery, amperage and voltage (SYS_STATUS) in MAV_DATA_STREAM_EXTENDED_STATUS
        •   - Gyro info (IMU_SCALED) in MAV_DATA_STREAM_EXTRA1
*/

if (!serial_has_space(FASTMAVLINK_MSG_REQUEST_DATA_STREAM_FRAME_LEN_MAX))
return 0;

fmav_msg_request_data_stream_encode_to_serial(my_sysid, my_compid, &payload, &status);

// request_data_stream_once = 0;

return 1;
}
// */
//############################################################################ END//request data stream
    command
```

```
//############################################################################ START//mission object input sequence command
//########################################## mavlink_msg_mission_count.h
// /*
uint8_t initiate_mission_object_input_sequence(void)
{
fmav_mission_count_t payload; // you could name the payload object inside here whatever you like

payload.count = 2;                // number of mission gps coordinates to upload !!! The first GPS coordinate is always the Home GPS position that is why we start the sequence counter from 2 and above
payload.target_system = my_sysid; // Ardupilot (autopilot) system id
payload.target_component = 0;     // Autopilot component id, 0 = all (seems to work fine)
payload.mission_type = 0;         //## 0: MAV_MISSION_TYPE_MISSION .Items are mission commands for main mission. //## 1: MAV_MISSION_TYPE_FENCE .Specifies GeoFence area(s). Items are MAV_CMD_NAV_FENCE_ GeoFence items. //## 2: MAV_MISSION_TYPE_RALLY .Specifies the rally points for the vehicle. Rally points are alternative RTL points. Items are MAV_CMD_NAV_RALLY_POINT rally point items. //## 255: MAV_MISSION_TYPE_ALL .Only used in MISSION_CLEAR_ALL to clear all mission types.

if (!serial_has_space(FASTMAVLINK_MSG_MISSION_COUNT_FRAME_LEN_MAX))
return 0;

fmav_msg_mission_count_encode_to_serial(my_sysid, my_compid, &payload, &status);

initiate_mission_once = 0;

return 1;
}
// */
//############################################################################ END//mission object input sequence command

//############################################################################ START//Home GPS position command
//########################################## mavlink_msg_mission_item.h
// /*
uint8_t create_home(int32_t Latitude, int32_t Longitude) // 32 bit integer number but the last 7 digits are the decimal points of the coordinate
{
fmav_mission_item_int_t payload; // you could name the payload object inside here whatever you like

payload.param1 = 0;                    // Loiter time
payload.param2 = 0;                    // Acceptable range from target - radius in meters
payload.param3 = 0;                    // Pass through waypoint
payload.param4 = 0;                    // Desired yaw angle
```

```
387.payload.x = Latitude;                   // Latitude - degrees
388.payload.y = Longitude;                  // Longitude - degrees
389.payload.z = 1;                          // Altitude (meters). This is irrelevant in a ground
    vehicle application
390.payload.seq = 0;                        // mission object Sequence number. Home mission item (rtl)
    is always the first mission item, so the sequence will be always "0"
391.payload.command = MAV_CMD_NAV_WAYPOINT; // command type
392.payload.target_system = my_sysid;       // Autopilot system id
393.payload.target_component = 0;           // Autopilot component id, 0 = all (seems to work fine)
394.payload.frame = 0;                      // Set target frame to global default
395.payload.current = 0;                    // false:0, true:1 - When downloading, whether the item is
    the current mission item.
396.payload.autocontinue = 0;               // Always 0
397.payload.mission_type = 0;               // should be the same as mission count //0: Items are
    mission commands for main mission. // 1: Specifies GeoFence area(s). Items are MAV_CMD_NAV_FENCE_
    GeoFence items. // 2: Specifies the rally points for the vehicle. Rally points are alternative RTL
    points. Items are MAV_CMD_NAV_RALLY_POINT rally point items. // 255: Only used in MISSION_CLEAR_ALL
    to clear all mission types.


398.if (!serial_has_space(FASTMAVLINK_MSG_MISSION_ITEM_INT_FRAME_LEN_MAX))
399.return 0;


400.fmav_msg_mission_item_int_encode_to_serial(my_sysid, my_compid, &payload, &status);


401.return 1;
402.}
403.// */
404.//########################################################################## END//Home GPS position
    command


405.//########################################################################## START//Waypoint GPS
    position command
406.//########################################### mavlink_msg_mission_item.h
407.// /*
408.uint8_t create_waypoint(int32_t Latitude, int32_t Longitude) // 32 bit integer number but the last
    7 digits are the decimal points of the coordinate
409.{


410.fmav_mission_item_int_t payload; // you could name the payload object inside here whatever you like


411.payload.param1 = 0;                      // Loiter time
412.payload.param2 = 0;                      // Acceptable range from target - radius in meters
413.payload.param3 = 0;                      // Pass through waypoint
414.payload.param4 = 0;                      // Desired yaw angle
415.payload.x = Latitude;                    // Latitude - degrees
416.payload.y = Longitude;                   // Longitude - degrees
```

```
417.payload.z = 1;                          // Altitude (meters). This is irrelevant in a ground
    vehicle application
418.payload.seq = 1;                        // mission object Sequence number
419.payload.command = MAV_CMD_NAV_WAYPOINT; // command type
420.payload.target_system = my_sysid;       // Autopilot system id
421.payload.target_component = 0;           // Autopilot component id, 0 = all (seems to work fine)
422.payload.frame = 0;                       // Set target frame to global default
423.payload.current = 1;                     // false:0, true:1 - When downloading, whether the item is
    the current mission item.
424.payload.autocontinue = 0;               // Always 0
425.payload.mission_type = 0;               // should be the same as mission count //0: Items are
    mission commands for main mission. // 1: Specifies GeoFence area(s). Items are MAV_CMD_NAV_FENCE_
    GeoFence items. // 2: Specifies the rally points for the vehicle. Rally points are alternative RTL
    points. Items are MAV_CMD_NAV_RALLY_POINT rally point items. // 255: Only used in MISSION_CLEAR_ALL
    to clear all mission types.


426.if (!serial_has_space(FASTMAVLINK_MSG_MISSION_ITEM_INT_FRAME_LEN_MAX))
427.return 0;


428.fmav_msg_mission_item_int_encode_to_serial(my_sysid, my_compid, &payload, &status);


429.return 1;
430.}
431.// */
432.//########################################################################## END//Waypoint GPS
    position command


433.//########################################################################## START//arm or disarm
    command
434.//########################################## mavlink_msg_command_long.h
435.// /*
436.uint8_t arm_disarm_autopilot(float arm_switch, float force_arm)
437.{
438.fmav_command_long_t payload; // you could name the payload object inside here whatever you like


439.payload.param1 = arm_switch;                  // 0: disarm, 1: arm
440.payload.param2 = force_arm;                   // 0: arm-disarm unless prevented by safety checks
    (i.e. when landed), 21196: force arming/disarming (e.g. allow arming to override preflight checks
    and disarming in flight)
441.payload.param3 = 0;                          // n/a
442.payload.param4 = 0;                          // n/a
443.payload.param5 = 0;                          // n/a
444.payload.param6 = 0;                          // n/a
445.payload.param7 = 0;                          // n/a
446.payload.command = MAV_CMD_COMPONENT_ARM_DISARM; // command type
447.payload.target_system = my_sysid;            // Autopilot system id
```

```
448.payload.target_component = 0;                  // Autopilot component id, 0 = all (seems to work
    fine)
449.payload.confirmation = 0;                      // Set confirmation message

450.if (!serial_has_space(FASTMAVLINK_MSG_COMMAND_LONG_FRAME_LEN_MAX))
451.return 0;

452.fmav_msg_command_long_encode_to_serial(my_sysid, my_compid, &payload, &status);

453.arm_disarm_once = 0;

454.return 1;
455.}
456.// */
457.//############################################################################ END//arm or disarm
    command

458.//############################################################################ START//command return to
    home
459.//########################################### mavlink_msg_command_long.h
460.// /*
461.uint8_t return_to_launch_autopilot(void) //(rtl or RTL = RETURN TO LAUNCH) (Home = Launch, position
    coordinates)
462.{
463.fmav_command_long_t payload; // you could name the payload object inside here whatever you like

464.payload.param1 = 0;                              // n/a
465.payload.param2 = 0;                              // n/a
466.payload.param3 = 0;                              // n/a
467.payload.param4 = 0;                              // n/a
468.payload.param5 = 0;                              // n/a
469.payload.param6 = 0;                              // n/a
470.payload.param7 = 0;                              // n/a
471.payload.command = MAV_CMD_NAV_RETURN_TO_LAUNCH; // command type
472.payload.target_system = my_sysid;                // Autopilot system id
473.payload.target_component = 0;                   // Autopilot component id, 0 = all (seems to work
    fine)
474.payload.confirmation = 0;                       // Set confirmation message

475.if (!serial_has_space(FASTMAVLINK_MSG_COMMAND_LONG_FRAME_LEN_MAX))
476.return 0;

477.fmav_msg_command_long_encode_to_serial(my_sysid, my_compid, &payload, &status);
```

```
478.rtl_once = 0;

479.client.publish("autopilot/mission/item_reached", "returning to home", false, 2); //("topic",
    "message_payload", boolean: retain_flag, integer: qos)

480.return 1;
481.}
482.// */
483.//########################################################################## END//command return to
    home

484.//#################################################################################################
    START//command request home position coordinates
485.//########################################## mavlink_msg_command_long.h
486.// /*
487.uint8_t request_rtl_coordinates(void) //(rtl or RTL = RETURN TO LAUNCH) (Home = Launch, position
    coordinates)
488.{
489.fmav_command_long_t payload; // you could name the payload object inside here whatever you like

490.payload.param1 = 0;                          // n/a
491.payload.param2 = 0;                          // n/a
492.payload.param3 = 0;                          // n/a
493.payload.param4 = 0;                          // n/a
494.payload.param5 = 0;                          // n/a
495.payload.param6 = 0;                          // n/a
496.payload.param7 = 0;                          // n/a
497.payload.command = MAV_CMD_GET_HOME_POSITION; // command type
498.payload.target_system = my_sysid;            // Autopilot system id
499.payload.target_component = 0;                // Autopilot component id, 0 = all (seems to work
    fine)
500.payload.confirmation = 0;                   // Set confirmation message

501.if (!serial_has_space(FASTMAVLINK_MSG_COMMAND_LONG_FRAME_LEN_MAX))
502.return 0;

503.fmav_msg_command_long_encode_to_serial(my_sysid, my_compid, &payload, &status);

504.request_rtl_coordinates_once = 0;

505.return 1;
506.}
507.// */
508.//#################################################################################################
    END//command request home position coordinates
```

```
509.//################################################################################
    START//command set home position coordinates
510.//########################################## mavlink_msg_command_int.h  (new method)
511.// /*
512.uint8_t set_home_gps_coordinates(int32_t Latitude, int32_t Longitude) // 32 bit integer number but
    the last 7 digits are the decimal points of the coordinate
513.{
514.fmav_command_int_t payload; // you could name the payload object inside here whatever you like

515.payload.param1 = 0;                      // (1=use current location, 0=use specified location)
516.payload.param2 = 0;                      // n/a
517.payload.param3 = 0;                      // n/a
518.payload.param4 = 0;                      // Yaw angle. NaN to use default heading
519.payload.x = Latitude;                    // latitude coordinate
520.payload.y = Longitude;                   // longitude coordinate
521.payload.z = 1;                           // Altitude (meters). This is irrelevant in a ground
    vehicle application
522.payload.command = MAV_CMD_DO_SET_HOME;  // command type
523.payload.target_system = my_sysid;        // Autopilot system id
524.payload.target_component = 0;           // Autopilot component id, 0 = all (seems to work fine)
525.payload.frame = 0;                      // Set target frame to global default
526.payload.current = 0;                    // false:0, true:1 - When downloading, whether the item is
    the current mission item.
527.payload.autocontinue = 0;               // Always 0

528.if (!serial_has_space(FASTMAVLINK_MSG_COMMAND_INT_FRAME_LEN_MAX))
529.return 0;

530.fmav_msg_command_int_encode_to_serial(my_sysid, my_compid, &payload, &status);

531.set_home_gps_coordinates_once = 0;

532.return 1;
533.}
534.// */
535.//################################################################################
    END//command set home position coordinates

536.//################################################################################
    START//command send heartbeat
537.// send a HEARTBEAT message
538.// returns 1 if successful, 0 else
539.uint8_t sendHeartbeat(void)
540.{
```

```
541.// Here we don't use the payload structure and xxx_encode() function, but rather pack the data
542.// directly into the serials' transmit buffer using the xxx_pack_to_serial() function. Just to
543.// show also this. This is however by no really much better RAM wise than the technique used in
544.// the "better" example, since a payload structure is put on stack inside the function.

545.if (!serial_has_space(FASTMAVLINK_MSG_HEARTBEAT_FRAME_LEN_MAX))
546.return 0;

547.fmav_msg_heartbeat_pack_to_serial(
548.my_sysid, my_compid,
549.MAV_TYPE_GENERIC, MAV_AUTOPILOT_INVALID, MAV_MODE_FLAG_SAFETY_ARMED, 0, MAV_STATE_ACTIVE,
550.&status);

551.return 1; // we have successfully send it, so tell that
552.}
553.//#############################################################################################
    END//command send heartbeat

554.//#############################################################################################
    START//MAVLINK message handler
555.// our message handler
556.void handleMessage(fmav_message_t *msg)
557.{
558.switch (msg->msgid)
559.{
560.case FASTMAVLINK_MSG_ID_HEARTBEAT:
561.{

562.// We use here the payload structure and the xxx_decode() function.
563.// This can stress the stack, since for some messages the payload can be large.
564.// Despite that it often might be the best option. Here we shall be fine with it.

565.fmav_heartbeat_t payload; // you could name the payload object inside here whatever you like
566.fmav_msg_heartbeat_decode(&payload, msg);

567.#if (enable_debug_serial == true)
568.// Serial.print(millis());
569.Serial.print("\nFlight Mode: (10 is auto)");
570.Serial.println(payload.custom_mode); ///< A bitfield for use for autopilot-specific flags.
571.Serial.print("Vehicle Type: ");
572.Serial.println(payload.type); ///< Type of the MAV (quadrotor, helicopter, etc., up to 15 types,
    defined in MAV_TYPE ENUM)
573.Serial.print("Autopilot type: ");
574.Serial.println(payload.autopilot); ///< Autopilot type / class. defined in MAV_AUTOPILOT ENUM
575.Serial.print("Base Mode: ");
```

```
576.Serial.println(payload.base_mode); ///< System mode bitfield, see MAV_MODE_FLAGS ENUM in
    mavlink/include/mavlink_types.h
577.Serial.print("System Status: ");
578.Serial.println(payload.system_status); ///< System status flag, see MAV_STATE ENUM (Ready to fly,
    Armed, Disarmed etc)
579.Serial.print("Mavlink Version: ");
580.Serial.println(payload.mavlink_version); ///< MAVLink version, not writable by user, gets added by
    protocol because of magic data type: uint8_t_mavlink_version
581.// Serial.println();
582.#endif

583.// Here you now can do something with the data in the payload.
584.// We do this: When we observe the autopilot, we simply trigger the emission of
585.// a "Hello World" statustext message.

586.client.publish("autopilot/vehicle/type", String(payload.type).c_str(), false,
    2);              //("topic", "message_payload", boolean: retain_flag, integer: qos)
587.client.publish("autopilot/vehicle/flightmode", String(payload.custom_mode).c_str(), false, 2);
    //("topic", "message_payload", boolean: retain_flag, integer: qos)
588.client.publish("autopilot/vehicle/status", String(payload.system_status).c_str(), false,
    2);   //("topic", "message_payload", boolean: retain_flag, integer: qos)

589.// This is a major flaw of MAVLink, there is no defined unambiguous procedure for
590.// identifying the nature of a component. So we do some simple heuristics here
591.// (which may be too simplistic for more realistic applications).

592.// un-comment this if you want to detect the flight controller
593.if (payload.autopilot != MAV_AUTOPILOT_INVALID && msg->compid == MAV_COMP_ID_AUTOPILOT1)
594.{
595.// un-comment this if you want to detect the GCS
596.// if (payload.autopilot == MAV_AUTOPILOT_INVALID && payload.type == MAV_TYPE_GCS) {
597.//  send_statustext = 1;
598.}
599.}
600.break;

601.case FASTMAVLINK_MSG_ID_GPS_RAW_INT:
602.{ // mavlink_msg_gps_raw_int.h

603.fmav_gps_raw_int_t payload; // you could name the payload object inside here whatever you like
604.fmav_msg_gps_raw_int_decode(&payload, msg);

605.#if (enable_debug_serial == true)
606.Serial.print("\nGPS Fix: ");
607.Serial.println(payload.fix_type);
```

```
608.Serial.print("GPS Latitude: ");
609.Serial.println(payload.lat);
610.Serial.print("GPS Longitude: ");
611.Serial.println(payload.lon);
612.Serial.print("GPS Speed: ");
613.Serial.println(payload.vel);
614.Serial.print("Sats Visible: ");
615.Serial.println(payload.satellites_visible);
616.// Serial.print("System Timestamp: ");Serial.println(payload.time_usec);
617.#endif

618.// client.publish("autopilot/gps/timestamp", uint64ToString(payload.time_usec), false, 2);
    //("topic", "message_payload", boolean: retain_flag, integer: qos)
619.client.publish("autopilot/gps/fix_type", String(payload.fix_type).c_str(), false,
    2);                    //("topic", "message_payload", boolean: retain_flag, integer: qos)
620.client.publish("autopilot/gps/speed", String(payload.vel).c_str(), false,
    2);                            //("topic", "message_payload", boolean: retain_flag, integer: qos)
621.client.publish("autopilot/gps/satelites_visible", String(payload.satellites_visible).c_str(),
    false, 2); //("topic", "message_payload", boolean: retain_flag, integer: qos)

622.String latitude_coord = String(payload.lat).c_str();
623.String longitude_coord = String(payload.lon).c_str();
624.// String sys_timestamp = uint64ToString(payload.time_usec); // uint64ToString
625.String latitude_name = "latitude";
626.String longitude_name = "longitude";
627.// String timestamp_name = "timestamp";
628.String bracket1 = "{";
629.String bracket2 = "}";
630.char comma = ',';
631.char ear = '"';
632.char doubledot = ':';

633.// gps_raw_data = bracket1 + ear + latitude_name + ear + doubledot + latitude_coord + comma + ear +
    longitude_name + ear + doubledot + longitude_coord + comma + ear + timestamp_name + ear + doubledot
    + sys_timestamp + bracket2;
634.gps_raw_data = bracket1 + ear + latitude_name + ear + doubledot + latitude_coord + comma + ear +
    longitude_name + ear + doubledot + longitude_coord + bracket2;

635.client.publish("autopilot/gps/coordinates", gps_raw_data, false, 2); //("topic", "message_payload",
    boolean: retain_flag, integer: qos) //SEND JSON
636.}
637.break;

638.case FASTMAVLINK_MSG_ID_SYS_STATUS:
639.{ // mavlink_msg_sys_status.h
```

```
640.fmav_sys_status_t payload; // you could name the payload object inside here whatever you like
641.fmav_msg_sys_status_decode(&payload, msg);

642.#if (enable_debug_serial == true)
643.Serial.print("\nAutopilot CPU load: ");
644.Serial.println(payload.load);
645.Serial.print("Battery Voltage: ");
646.Serial.println(payload.voltage_battery);
647.Serial.print("Battery Current: ");
648.Serial.println(payload.current_battery);
649.Serial.print("Battery Capacity Remaining: ");
650.Serial.println(payload.battery_remaining);
651.Serial.print("Communications packetloss: ");
652.Serial.println(payload.drop_rate_comm);
653.#endif

654.client.publish("autopilot/cpu/load", String(payload.load).c_str(), false,
    2);                                        //("topic", "message_payload", boolean: retain_flag,
    integer: qos)
655.client.publish("autopilot/vehicle/battery/voltage", String(payload.voltage_battery).c_str(), false,
    2);              //("topic", "message_payload", boolean: retain_flag, integer: qos)
656.client.publish("autopilot/vehicle/battery/current", String(payload.current_battery).c_str(), false,
    2);              //("topic", "message_payload", boolean: retain_flag, integer: qos)
657.client.publish("autopilot/vehicle/battery/capacity_remaining",
    String(payload.battery_remaining).c_str(), false, 2); //("topic", "message_payload", boolean:
    retain_flag, integer: qos)
658.client.publish("autopilot/communications/packetloss", String(payload.drop_rate_comm).c_str(),
    false, 2);             //("topic", "message_payload", boolean: retain_flag, integer: qos)
659.}
660.break;

661.case FASTMAVLINK_MSG_ID_MISSION_REQUEST:
662.{ // mavlink_msg_mission_request.h

663.// uploading a new waypoint - Check for mission request from autopilot after the
    "initiate_mission_object_input_sequence()" function was executed and continue by sending the
    mission GPS waypoints
664.fmav_mission_request_t payload; // you could name the payload object inside here whatever you like
665.fmav_msg_mission_request_decode(&payload, msg);

666.#if (enable_debug_serial == true)
667.Serial.print("\nMission Req Sequence: ");
668.Serial.println(payload.seq);
669.Serial.print("\SysID: ");
670.Serial.println(payload.target_system);
671.Serial.print("\Compid: ");
```

```
672.Serial.println(payload.target_component);
673.#endif

674.if (payload.seq == 0)
675.{ // checking payload object sequence request number
676.create_home(rtl_cmd_latitude, rtl_cmd_longitude);
677.#if (enable_debug_serial == true)
678.Serial.print("Sent Home: \n");
679.#endif
680.}

681.if (payload.seq == 1)
682.{ // checking payload object sequence request number
683.create_waypoint(mission_cmd_latitude, mission_cmd_longitude);
684.#if (enable_debug_serial == true)
685.Serial.print("Sent Waypoint: \n");
686.#endif

687.String mission_lat = String(mission_cmd_latitude).c_str();
688.String mission_long = String(mission_cmd_longitude).c_str();
689.// String sys_timestamp = uint64ToString(payload.time_usec); // uint64ToString
690.String latitude_name = "latitude";
691.String longitude_name = "longitude";
692.// String timestamp_name = "timestamp";
693.String bracket1 = "{";
694.String bracket2 = "}";
695.char comma = ',';
696.char ear = '"';
697.char doubledot = ':';

698.mission_gps_data = bracket1 + ear + latitude_name + ear + doubledot + mission_lat + comma + ear +
    longitude_name + ear + doubledot + mission_long + bracket2;

699.client.publish("autopilot/mission/coordinates", mission_gps_data, false, 2); //("topic",
    "message_payload", boolean: retain_flag, integer: qos) //SEND JSON

700.client.publish("autopilot/mission/item_reached", "mission in progress", false, 2); //("topic",
    "message_payload", boolean: retain_flag, integer: qos)
701.}
702.}
703.break;

704.case FASTMAVLINK_MSG_ID_MISSION_ACK:
705.{ // mavlink_msg_mission_ack.h
```

```
706.// uploading a new waypoint - Receive Mission Ack Message from autopilot
707.fmav_mission_ack_t payload; // you could name the payload object inside here whatever you like
708.fmav_msg_mission_ack_decode(&payload, msg);


709.#if (enable_debug_serial == true)
710.Serial.print("\nMission Ack Sequence: ");
711.Serial.println(payload.type);
712.Serial.print("\SysID: ");
713.Serial.println(payload.target_system);
714.Serial.print("\CompID: ");
715.Serial.println(payload.target_component);


716.if (payload.type == 1)
717.{
718.Serial.print("\nMission upload FAILED: ");
719.Serial.println(payload.type);
720.}


721.if (payload.type == 0)
722.{
723.Serial.print("\nMission upload SUCCESSFULL: ");
724.Serial.println(payload.type);
725.}


726.Serial.print("type of mission: ");
727.if (payload.mission_type == 0)
728.{
729.Serial.println("Main Mission Coordinates");
730.}
731.else if (payload.mission_type == 1)
732.{
733.Serial.println("Mission Geofence Coordinates");
734.}
735.else if (payload.mission_type == 2)
736.{
737.Serial.println("Mission Rally point Coordinates");
738.}
739.else if (payload.mission_type == 255)
740.{
741.Serial.println("Mission Clear Coordinates");
742.}
743.#endif


744.String ack_type = String(payload.type).c_str();
745.String mission_type = String(payload.mission_type).c_str();
746.String system_id = String(payload.target_system).c_str();
```

```
String component_id = String(payload.target_component).c_str();
String ack_name = "acknowledge type";
String mission_name = "mission type";
String sysname = "system id";
String compname = "component id";
String bracket1 = "{";
String bracket2 = "}";
char comma = ',';
char ear = '"';
char doubledot = ':';

mission_ack_data = bracket1 + ear + ack_name + ear + doubledot + ack_type + comma + ear +
    mission_name + ear + doubledot + mission_type + comma + ear + sysname + ear + doubledot + system_id
    + comma + ear + compname + ear + doubledot + component_id + bracket2;

client.publish("autopilot/mission/ack", mission_ack_data, false, 2); //("topic", "message_payload",
    boolean: retain_flag, integer: qos) //SEND JSON
}
break;

case FASTMAVLINK_MSG_ID_COMMAND_ACK:
{ // mavlink_msg_command_ack.h

// receive the command acknowledge message
fmav_command_ack_t payload; // you could name the payload object inside here whatever you like
fmav_msg_command_ack_decode(&payload, msg);

#if (enable_debug_serial == true)
Serial.print("\nCommand: ");
Serial.println(payload.command);
Serial.print("command result: ");
Serial.println(payload.result);
Serial.print("progress: ");
Serial.println(payload.progress);
Serial.print("result of param 2: ");
Serial.println(payload.result_param2);
Serial.print("target system: ");
Serial.println(payload.target_system);
Serial.print("target component: ");
Serial.println(payload.target_component);
#endif

String command_code = String(payload.command).c_str();
String command_result = String(payload.result).c_str();
String target_sys = String(payload.target_system).c_str();
String target_comp = String(payload.target_component).c_str();
```

```
String command_name = "command";
String result_name = "result";
String sysname = "target system id";
String compname = "target component id";
String bracket1 = "{";
String bracket2 = "}";
char comma = ',';
char ear = '"';
char doubledot = ':';

command_ack_data = bracket1 + ear + command_name + ear + doubledot + command_code + comma + ear +
    result_name + ear + doubledot + command_result + comma + ear + sysname + ear + doubledot +
    target_sys + comma + ear + compname + ear + doubledot + target_comp + bracket2;

client.publish("autopilot/command/ack", command_ack_data, false, 2); //("topic", "message_payload",
    boolean: retain_flag, integer: qos) //SEND JSON
}
break;

case FASTMAVLINK_MSG_ID_MISSION_ITEM_REACHED:
{ // mavlink_msg_mission_item_reached.h

// uploading the arm command
fmav_mission_item_reached_t payload; // you could name the payload object inside here whatever you
    like
fmav_msg_mission_item_reached_decode(&payload, msg);

#if (enable_debug_serial == true)
Serial.print("\nMission item reached: ");
Serial.println(payload.seq); // should show "2" if destination reached
#endif

client.publish("autopilot/mission/item_reached", "mission reached", false, 2); //("topic",
    "message_payload", boolean: retain_flag, integer: qos)
}
break;

case FASTMAVLINK_MSG_ID_HOME_POSITION:
{ // mavlink_msg_home_position.h

// receive the return to launch gps coordinates
fmav_home_position_t payload; // you could name the payload object inside here whatever you like
fmav_msg_home_position_decode(&payload, msg);
```

```
814.#if (enable_debug_serial == true)
815.Serial.print("\nHome potition received: ");
816.Serial.println("");
817.Serial.print("Latitude: ");
818.Serial.println(payload.latitude);
819.Serial.print("Longitude: ");
820.Serial.println(payload.longitude);
821.Serial.print("Altitude: ");
822.Serial.println(payload.altitude);
823.Serial.print("Local coordinate x: ");
824.Serial.println(payload.x);
825.Serial.print("Local coordinate y: ");
826.Serial.println(payload.y);
827.Serial.print("Local coordinate z: ");
828.Serial.println(payload.z);
829.Serial.print("Heading and Ground slope q: ");
830.Serial.print(payload.q[0]);
831.Serial.print(", ");
832.Serial.print(payload.q[1]);
833.Serial.print(", ");
834.Serial.print(payload.q[2]);
835.Serial.print(", ");
836.Serial.println(payload.q[3]);
837.Serial.print("Local coordinate approach_x: ");
838.Serial.println(payload.approach_x);
839.Serial.print("Local coordinate approach_y: ");
840.Serial.println(payload.approach_y);
841.Serial.print("Local coordinate approach_z: ");
842.Serial.println(payload.approach_z);
843.// Serial.print("System Timestamp: ");Serial.println(payload.time_usec);
844.#endif


845.String rtl_lat = String(payload.latitude).c_str();
846.String rtl_long = String(payload.longitude).c_str();
847.// String sys_timestamp = uint64ToString(payload.time_usec); // uint64ToString
848.String latitude_name = "latitude";
849.String longitude_name = "longitude";
850.// String timestamp_name = "timestamp";
851.String bracket1 = "{";
852.String bracket2 = "}";
853.char comma = ',';
854.char ear = '"';
855.char doubledot = ':';


856.// gps_rtl_data = bracket1 + ear + latitude_name + ear + doubledot + rtl_lat + comma + ear +
    longitude_name + ear + doubledot + rtl_long + comma + ear + timestamp_name + ear + doubledot +
    sys_timestamp + bracket2;
```

```
857.gps_rtl_data = bracket1 + ear + latitude_name + ear + doubledot + rtl_lat + comma + ear +
    longitude_name + ear + doubledot + rtl_long + bracket2;

858.client.publish("autopilot/gps/rtl_coordinates", gps_rtl_data, false, 2); //("topic",
    "message_payload", boolean: retain_flag, integer: qos) //SEND JSON
859.}
860.break;

861.} // end of switch
862.} // end of "void handleMessage(fmav_message_t* msg)"
863.//#############################################################################################
    END//MAVLINK message handler

864.//#############################################################################################
    START//MAVLINK command handler
865.// this should be called repeatedly
866.void spinOnce(void)
867.{
868.// let's first check and do incoming messages
869.uint16_t available = serial_available();
870.for (uint16_t i = 0; i < available; i++)
871.{
872.uint8_t c;
873.serial_read_char(&c);

874.uint8_t res = fmav_parse_to_msg(&msg, &status, c);

875.if (res == FASTMAVLINK_PARSE_RESULT_OK)
876.{
877.if (fmav_msg_is_for_me(my_sysid, my_compid, &msg))
878.{
879.handleMessage(&msg);
880.}
881.}
882.}

883.// lets request a data stream from the autopilot
884.if (request_data_stream_once == 1)
885.{ // if the request of data stream flag is activated, enable the stream once and then dont send the
    same request again
886.sendRequestDataStream(1);
887.}

888.// lets send a mission to the autopilot
889.if (initiate_mission_once == 1)
```

```
{ // if the request of a mission flag is activated, then "initiate_mission_object_input_sequence()" once and then dont send the same request again
initiate_mission_object_input_sequence();
}

// lets send a command to the autopilot: ARM/DISARM
if (arm_disarm_once == 1)
{                                           // if the request of a mission flag is activated, then "arm_disarm_autopilot(x,y)" once and then dont send the same request again
arm_disarm_autopilot(arm_disarm_switch, 0); // for the arm_disarm switch  Arm: 0 or 1 ,Force: 0 or 21196
}

// lets send a command to the autopilot: request the home position GPS coordinates
if (request_rtl_coordinates_once == 1)
{ // if the request of a mission flag is activated, then "request_rtl_coordinates()" once and then dont send the same request again
request_rtl_coordinates();
}

// lets send a command to the autopilot: set new home position GPS coordinates
if (set_home_gps_coordinates_once == 1)
{ // if the request of a mission flag is activated, then "set_home_gps_coordinates()" once and then dont send the same request again
// set_home_gps_coordinates(0,rtl_cmd_latitude,rtl_cmd_longitude,1); // (1=use current location, 0=use specified location), (float use_current_position_flag, float Latitude, float Longitude , float Altitude)
set_home_gps_coordinates(rtl_cmd_latitude, rtl_cmd_longitude); // (1=use current location, 0=use specified location), (float use_current_position_flag, float Latitude, float Longitude , float Altitude)
}

if (rtl_once == 1)
{ //if the return to launch flag is activated, then call the "request_to_launch_autopilot()" message function
return_to_launch_autopilot();
}

// let's now send a HEARTBEAT message, to tell the world we exist
uint32_t tnow_ms = get_time_ms();
if ((tnow_ms - tlast_ms) > 1000)
{
if (sendHeartbeat())
{
tlast_ms += 1000; // we have successfully send it, so do it again later
LED_TOGGLE;
```

```
921.}
922.}
923.}
924.//################################################################################
    END//MAVLINK command handler

925.//------------------------------
926.// Default Arduino setup() and loop() functions
927.//------------------------------

928.void setup()
929.{
930.//################################ START//INITIALISING SERIAL PORTS
931.Mavlink_Serial.begin(SERIAL_BAUD, SERIAL_8N1, RXD2, TXD2); // hardware serial
932.// Mavlink_Serial.begin(57600, SWSERIAL_8N1, swRX, swTX, false, 256); //software serial

933.// SERIAL.begin(SERIAL_BAUD); //old mavlink serial port object

934.#if (enable_debug_serial == true)
935.Serial.begin(SERIAL_BAUD); // Main serial port for console output

936.Serial.println(" ");
937.Serial.println("..................................................................................."
    );
938.Serial.println("..................................................................................."
    );
939.Serial.println("..................................................................................."
    );
940.Serial.println("..................................................................................."
    );
941.Serial.println("..................................................................................."
    );
942.Serial.println(" ");
943.Serial.println("System Ready...");
944.#endif
945.//################################ END//INITIALISING SERIAL PORTS

946.//################################ START//MQTT + WIFI SETUP
947.WiFi.begin(ssid, pass);

948.// Note: Local domain names (e.g. "Computer.local" on OSX) are not supported
949.// by Arduino. You need to set the IP address directly.
950.client.begin("83.212.77.23", 1883, net); // ("mqtt_broker_ip", "network_object_variable")
951.client.onMessage(messageReceived);
```

```
952.connect();
953.//################################ END//MQTT + WIFI SETUP


954.pinMode(BLINK_PIN, OUTPUT);


955.fmav_init(); // let's always call it, even if it currently may not do anything
956.}


957.void loop()
958.{
959.//################################ START//MQTT + WIFI
960.client.loop();
961.delay(10); // <- fixes some issues with WiFi stability. Make it "20" for esp8266


962.if (!client.connected())
963.{
964.connect();
965.}


966.//################################ START//MAVLINK command handler
967.spinOnce();
968.//################################ END//MAVLINK command handler


969.// publish a message roughly every second.
970.if (millis() - lastMillis > 1000)
971.{
972.lastMillis = millis();
973.// boolean retained = false;
974.// int qos = 2;
975.// client.publish("/hello", "world", retained, qos); //("topic", "message_payload")
976.client.publish("autopilot/companion_computer/RSSI", String(WiFi.RSSI()).c_str(), false, 2);
    //("topic", "message_payload", boolean: retain_flag, integer: qos)
977.// mqtt data publish functions moved to mavlink "handleMessage" functions
978.}
979.//################################ END//MQTT + WIFI
980.}
```

Παράρτημα Γ: Κώδικας ροών της πλατφόρμας Node-RED.


```
[{"id":"a7681dcb045c877f","type":"tab","label":"BOAT","disabled":false,"info":""},{"id":"fb33f5a87ea6fa
a8","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/command/ack","qos":"2","datatype":"auto","broke
r":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":300,"y":280,"wires":[["c09bdfae29a67e
e7","cb9bdfa55c4332df"]]},{"id":"149e9a73a02c2c19","type":"debug","z":"a7681dcb045c877f","name":"From
autopilot
general","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetT
ype":"msg","statusVal":"","statusType":"auto","x":700,"y":300,"wires":[]},{"id":"c09bdfae29a67ee7","typ
e":"json","z":"a7681dcb045c877f","name":"","property":"payload","action":"","pretty":false,"x":650,"y":
400,"wires":[["149e9a73a02c2c19","26400ad8bf97cad3"]]},{"id":"6a1d068a6e7c45fb","type":"ui_template","z
":"a7681dcb045c877f","group":"0f1f8234940685b4","name":"Command
ACK","order":2,"width":6,"height":"3","format":"<p><span style=\"color: #003366;\"><em><span
style=\"text-decoration: underline;\"><strong>Command ACK</strong></span></em></span></p>\n<table>\n
<tr ng-repeat=\"el in msg.payload\">\n        <td><b>{{el[0]}}</b></td>\n        <td>{{el[1]}}</td>\n
</tr>\n\n</table>","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":
"local","className":"","x":1580,"y":160,"wires":[[]],"inputLabels":["json
message"]},{"id":"55f3b370ff50e05b","type":"function","z":"a7681dcb045c877f","name":"Command
ACK","func":"msg.payload = Object.entries(msg.payload);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1240,"y":160,"wires":[["6a1d068
a6e7c45fb"]]},{"id":"9df01390fcb23c11","type":"ui_form","z":"a7681dcb045c877f","name":"","label":"Set
Gps Target
position","group":"cf03a5ffb3f477cd","order":2,"width":0,"height":0,"options":[{"label":"Enter
Latidute","value":"lat","type":"text","required":true,"rows":null},{"label":"Enter
Longitude","value":"lon","type":"text","required":true,"rows":null},{"label":"UNIX
timestamp","value":"Time","type":"checkbox","required":true,"rows":null}],"formValue":{"lat":"","lon":"
","Time":false},"payload":"","submit":"submit","cancel":"cancel","topic":"topic","topicType":"msg","spl
itLayout":"","className":"","x":350,"y":2020,"wires":[["da54cd5f48357824"]]},{"id":"0d750d8c4fbbb8cd","
type":"mqtt
out","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/command/waypoint","qos":"2","retain":"
false","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"e8efa29ec5eec79
c","x":1270,"y":2020,"wires":[]},{"id":"ee86def5941debe1","type":"mqtt
out","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/command/request_rtl_coordinates","qos"
:"2","retain":"false","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"
e8efa29ec5eec79c","x":670,"y":2680,"wires":[]},{"id":"452ca11172cd9de3","type":"mqtt
out","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/command/arm_disarm","qos":"2","retain"
:"false","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"e8efa29ec5eec
79c","x":640,"y":2520,"wires":[]},{"id":"bf740b939cdc3ae8","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/gps/fix_type","qos":"2","datatype":"auto","brok
er":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":300,"y":780,"wires":[["d58f142839635
57c","630a02b5d2ea4bed"]]},{"id":"d58f14283963557c","type":"debug","z":"a7681dcb045c877f","name":"GPS
fix
type","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType
":"msg","statusVal":"","statusType":"auto","x":590,"y":740,"wires":[]},{"id":"26400ad8bf97cad3","type":
"switch","z":"a7681dcb045c877f","name":"","property":"topic","propertyType":"msg","rules":[{"t":"eq","v
":"autopilot/command/ack","vt":"str"},{"t":"eq","v":"autopilot/mission/ack","vt":"str"},{"t":"eq","v":"
```

```
autopilot/mission/coordinates","vt":"str"},{"t":"eq","v":"autopilot/gps/rtl_coordinates","vt":"str"},{"
t":"eq","v":"autopilot/gps/coordinates","vt":"str"}],"checkall":"true","repair":false,"outputs":5,"x":8
90,"y":400,"wires":[["55f3b370ff50e05b","20591289410e7a06"],["3b0f0bfe2d54ea7f","3bd54186f48dd8ba"],["b
054ee1abc6417b0","2d11c0c112188a94"],["65a269d0e5671f6b","82f00ee91c389291","93701550747b010f"],["6a64b
7bd6a978206","9396df79f8fce068","4aadec50314ffc2c"]]},{"id":"3b0f0bfe2d54ea7f","type":"debug","z":"a768
1dcb045c877f","name":"Mission
ACK","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":1230,"y":240,"wires":[]},{"id":"20591289410e7a06","type":
"debug","z":"a7681dcb045c877f","name":"Command
ACK","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":1240,"y":120,"wires":[]},{"id":"3bd54186f48dd8ba","type":
"function","z":"a7681dcb045c877f","name":"Mission ACK","func":"msg.payload =
Object.entries(msg.payload);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1230,"y":280,"wires":[["e16793a
e7b4cb8e6"]]},{"id":"e16793ae7b4cb8e6","type":"ui_template","z":"a7681dcb045c877f","group":"0f1f8234940
685b4","name":"Mission ACK","order":1,"width":6,"height":"3","format":"<p><span style=\"color:
#003366;\"><em><span style=\"text-decoration: underline;\"><strong>Mission
ACK</strong></span></em></span></p>\n<table>\n    <tr ng-repeat=\"el in msg.payload\">\n
<td><b>{{el[0]}}</b></td>\n        <td>{{el[1]}}</td>\n
</tr>\n\n</table>","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":
"local","className":"","x":1570,"y":280,"wires":[[]],"inputLabels":["json
message"]},{"id":"da54cd5f48357824","type":"function","z":"a7681dcb045c877f","name":"Add
Timestamp","func":"if (msg.payload.Time == true){\n   var d = new Date();\n   msg.payload.Time =
d.getTime();\n} else {\n    msg.payload.Time = \"False\"\n}\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":620,"y":2020,"wires":[["1a314e6
0261c35c5"]]},{"id":"b3fcb5c65629bf1b","type":"comment","z":"a7681dcb045c877f","name":"Receiving all
topics with JSON payload format","info":"Revceiving all
topics","x":410,"y":220,"wires":[]},{"id":"513ee616949f7102","type":"comment","z":"a7681dcb045c877f","n
ame":"Sets the missions
coordinates","info":"","x":340,"y":2060,"wires":[]},{"id":"714b061826a27f4c","type":"comment","z":"a768
1dcb045c877f","name":"Sends coordinates to autopilots specific
topic","info":"","x":1290,"y":1980,"wires":[]},{"id":"2d11c0c112188a94","type":"debug","z":"a7681dcb045
c877f","name":"mission
coordinatess","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","ta
rgetType":"msg","statusVal":"","statusType":"auto","x":1260,"y":360,"wires":[]},{"id":"2208c438dc1f9556
","type":"debug","z":"a7681dcb045c877f","name":"Waypoint
command","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetT
ype":"msg","statusVal":"","statusType":"auto","x":1210,"y":2080,"wires":[]},{"id":"1a314e60261c35c5","t
ype":"split","z":"a7681dcb045c877f","name":"","splt":"\\n","spltType":"str","arraySplt":1,"arraySpltTyp
e":"len","stream":false,"addname":"topic","x":790,"y":2020,"wires":[["bdb52fe87620748c"]]},{"id":"bdb52
fe87620748c","type":"join","z":"a7681dcb045c877f","name":"","mode":"custom","build":"string","property"
:"payload","propertyType":"msg","key":"topic","joiner":",","joinerType":"str","accumulate":false,"timeo
ut":"","count":"","reduceRight":false,"reduceExp":"","reduceInit":"","reduceInitType":"","reduceFixup":
"","x":930,"y":2020,"wires":[["2208c438dc1f9556","0d750d8c4fbbb8cd"]]},{"id":"9ec9b161c5f1a580","type":
"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/ack","qos":"2","datatype":"auto","broke
r":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":310,"y":340,"wires":[["c09bdfae29a67e
e7"]]},{"id":"26a713f59680baf0","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/coordinates","qos":"2","datatype":"auto
```

```
","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":280,"y":400,"wires":[["c09bdf
ae29a67ee7"]]},{"id":"10748ef19fa5de89","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/gps/rtl_coordinates","qos":"2","datatype":"auto
","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":280,"y":460,"wires":[["c09bdf
ae29a67ee7"]]},{"id":"b054ee1abc6417b0","type":"function","z":"a7681dcb045c877f","name":"mission
coordinates","func":"msg.payload = Object.entries(msg.payload);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1260,"y":400,"wires":[["b1b82c0
dd0219ab1"]]},{"id":"65a269d0e5671f6b","type":"function","z":"a7681dcb045c877f","name":"gps rtl
coordinates","func":"msg.payload = Object.entries(msg.payload);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1250,"y":520,"wires":[["bd664dd
72ab74410"]]},{"id":"b1b82c0dd0219ab1","type":"ui_template","z":"a7681dcb045c877f","group":"cf03a5ffb3f
477cd","name":"Mission coordinates","order":3,"width":6,"height":"3","format":"<p><span style=\"color:
#003366;\"><em><span style=\"text-decoration: underline;\"><strong>Mission
coordinates</strong></span></em></span></p>\n<table>\n    <tr ng-repeat=\"el in msg.payload\">\n
<td><b>{{el[0]}}</b></td>\n        <td>{{el[1]}}</td>\n
</tr>\n\n</table>","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":
"local","className":"","x":1600,"y":400,"wires":[[]],"inputLabels":["json
message"]},{"id":"bd664dd72ab74410","type":"ui_template","z":"a7681dcb045c877f","group":"8afb3977543fc7
ab","name":"GPS RTL coordinates","order":4,"width":6,"height":"3","format":"<p><span style=\"color:
#003366;\"><em><span style=\"text-decoration: underline;\"><strong>GPS RTL
coordinates</strong></span></em></span></p>\n<table>\n    <tr ng-repeat=\"el in msg.payload\">\n
<td><b>{{el[0]}}</b></td>\n        <td>{{el[1]}}</td>\n
</tr>\n\n</table>","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":
"local","className":"","x":1600,"y":520,"wires":[[]],"inputLabels":["json
message"]},{"id":"82f00ee91c389291","type":"debug","z":"a7681dcb045c877f","name":"gps rtl
coordinates","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","tar
getType":"msg","statusVal":"","statusType":"auto","x":1250,"y":480,"wires":[]},{"id":"99a53d86587e4d1d"
,"type":"ui_form","z":"a7681dcb045c877f","name":"","label":"Set RTL GPS
position","group":"8afb3977543fc7ab","order":3,"width":0,"height":0,"options":[{"label":"Enter
Latidute","value":"lat","type":"text","required":true,"rows":null},{"label":"Enter
Longitude","value":"lon","type":"text","required":true,"rows":null},{"label":"UNIX
timestamp","value":"Time","type":"checkbox","required":true,"rows":null}],"formValue":{"lat":"","lon":"
","Time":false},"payload":"","submit":"submit","cancel":"cancel","topic":"topic","topicType":"msg","spl
itLayout":"","className":"","x":360,"y":2360,"wires":[["34a09a55928afd9e"]]},{"id":"34a09a55928afd9e","
type":"function","z":"a7681dcb045c877f","name":"Add Timestamp","func":"if (msg.payload.Time == true){\n
var d = new Date();\n   msg.payload.Time = d.getTime();\n} else {\n    msg.payload.Time =
\"False\"\n}\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":620,"y":2360,"wires":[["aeb3b38
739257d94"]]},{"id":"151a67970fa5da05","type":"debug","z":"a7681dcb045c877f","name":"RTL
commmand","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","target
Type":"msg","statusVal":"","statusType":"auto","x":1200,"y":2420,"wires":[]},{"id":"aeb3b38739257d94","
type":"split","z":"a7681dcb045c877f","name":"","splt":"\\n","spltType":"str","arraySplt":1,"arraySpltTy
pe":"len","stream":false,"addname":"topic","x":790,"y":2360,"wires":[["6f3e354dae63b3ec"]]},{"id":"6f3e
354dae63b3ec","type":"join","z":"a7681dcb045c877f","name":"","mode":"custom","build":"string","property
":"payload","propertyType":"msg","key":"topic","joiner":",","joinerType":"str","accumulate":false,"time
out":"","count":"","reduceRight":false,"reduceExp":"","reduceInit":"","reduceInitType":"","reduceFixup"
:"","x":930,"y":2360,"wires":[["151a67970fa5da05","c77345464ffb9c39"]]},{"id":"c77345464ffb9c39","type"
:"mqtt
out","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/command/set_rtl_coordinates","qos":"2"
```

```
,"retain":"false","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"e8ef
a29ec5eec79c","x":1300,"y":2360,"wires":[]},{"id":"56b8896960ef16de","type":"ui_switch","z":"a7681dcb04
5c877f","name":"","label":"Arm/Disarm
switch","tooltip":"","group":"c245161a4a1c70c6","order":1,"width":0,"height":0,"passthru":true,"decoupl
e":"false","topic":"autopilot/mission/command/arm_disarm","topicType":"str","style":"","onvalue":"1","o
nvalueType":"str","onicon":"","oncolor":"","offvalue":"0","offvalueType":"str","officon":"","offcolor":
"","animate":false,"className":"","x":310,"y":2600,"wires":[["452ca11172cd9de3","3a1d814002e410d2"]]},{
"id":"3a1d814002e410d2","type":"debug","z":"a7681dcb045c877f","name":"Arm
switch","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetTy
pe":"msg","statusVal":"","statusType":"auto","x":550,"y":2600,"wires":[]},{"id":"4ed3bb5ef4b45c51","typ
e":"ui_button","z":"a7681dcb045c877f","name":"","group":"8afb3977543fc7ab","order":2,"width":0,"height"
:0,"passthru":false,"label":"Request RTL Coordinates
button","tooltip":"","color":"","bgcolor":"","className":"","icon":"","payload":"1","payloadType":"str"
,"topic":"topic","topicType":"msg","x":260,"y":2760,"wires":[["ee86def5941debe1","6f71918dfabdafc9"]]},
{"id":"6f71918dfabdafc9","type":"debug","z":"a7681dcb045c877f","name":"RTL Coordinates read
Button","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetTy
pe":"msg","statusVal":"","statusType":"auto","x":600,"y":2760,"wires":[]},{"id":"3bb66f7b1f6b1f7e","typ
e":"mqtt
out","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/command/return_to_launch","qos":"2","r
etain":"false","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"e8efa29
ec5eec79c","x":650,"y":2840,"wires":[]},{"id":"0e12ed3550a1f3f6","type":"ui_button","z":"a7681dcb045c87
7f","name":"","group":"8afb3977543fc7ab","order":5,"width":0,"height":0,"passthru":false,"label":"Retur
n to launch
button","tooltip":"","color":"","bgcolor":"","className":"","icon":"","payload":"1","payloadType":"str"
,"topic":"topic","topicType":"msg","x":290,"y":2920,"wires":[["3bb66f7b1f6b1f7e","6cf04c4fde2dd8cd"]]},
{"id":"6cf04c4fde2dd8cd","type":"debug","z":"a7681dcb045c877f","name":"Return to
Launch","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetTy
pe":"msg","statusVal":"","statusType":"auto","x":570,"y":2920,"wires":[]},{"id":"60442591b97c3668","typ
e":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/gps/coordinates","qos":"2","datatype":"auto","b
roker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":290,"y":520,"wires":[["c09bdfae29
a67ee7"]]},{"id":"6a64b7bd6a978206","type":"function","z":"a7681dcb045c877f","name":"gps current
coordinates","func":"msg.payload = Object.entries(msg.payload);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1270,"y":720,"wires":[["19231e3
d4c4cc4b8"]]},{"id":"19231e3d4c4cc4b8","type":"ui_template","z":"a7681dcb045c877f","group":"374b8b3839d
baf9e","name":"GPS current coordinates","order":2,"width":6,"height":"3","format":"<p><span
style=\"color: #003366;\"><em><span style=\"text-decoration: underline;\"><strong>GPS current
coordinates</strong></span></em></span></p>\n<table>\n    <tr ng-repeat=\"el in msg.payload\">\n
<td><b>{{el[0]}}</b></td>\n        <td>{{el[1]}}</td>\n
</tr>\n\n</table>","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":
"local","x":1610,"y":720,"wires":[[]],"inputLabels":["json
message"]},{"id":"4aadec50314ffc2c","type":"debug","z":"a7681dcb045c877f","name":"gps current
coordinates","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targ
etType":"msg","statusVal":"","statusType":"auto","x":1270,"y":680,"wires":[]},{"id":"f96c7962431083f8",
"type":"ui_text","z":"a7681dcb045c877f","group":"c245161a4a1c70c6","order":2,"width":"6","height":"2","
name":"","label":"GPS fix type","format":"{{msg.payload}}","layout":"col-
center","x":870,"y":780,"wires":[]},{"id":"3026acebed577803","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/gps/satelites_visible","qos":"2","datatype":"au
to","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":280,"y":900,"wires":[["e96a
```

```
f99a96598cc9","0365a7741ff5e781"]]},{"id":"e96af99a96598cc9","type":"debug","z":"a7681dcb045c877f","nam
e":"GPS satelites
visible","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetT
ype":"msg","statusVal":"","statusType":"auto","x":620,"y":860,"wires":[]},{"id":"0365a7741ff5e781","typ
e":"ui_text","z":"a7681dcb045c877f","group":"c245161a4a1c70c6","order":3,"width":0,"height":0,"name":""
,"label":"GPS satelites visible","format":"{{msg.payload}}","layout":"row-
spread","className":"","x":620,"y":900,"wires":[]},{"id":"3c38fec478b416a2","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/gps/speed","qos":"2","datatype":"auto","broker"
:"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":310,"y":1020,"wires":[["557971488251afb
d","f578a9ee5ef02558"]]},{"id":"557971488251afbd","type":"debug","z":"a7681dcb045c877f","name":"GPS
vehicle
speed","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetTyp
e":"msg","statusVal":"","statusType":"auto","x":610,"y":980,"wires":[]},{"id":"f578a9ee5ef02558","type"
:"ui_text","z":"a7681dcb045c877f","group":"374b8b3839dbaf9e","order":3,"width":"6","height":"2","name":
"","label":"GPS vehicle speed","format":"{{msg.payload}}  Km/h","layout":"row-
spread","className":"","x":610,"y":1020,"wires":[]},{"id":"3e9badaea0449f62","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/vehicle/battery/capacity_remaining","qos":"2","
datatype":"auto","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":230,"y":1140,"
wires":[["485b1efc786da48b","f57882ebd6dc0c9a"]]},{"id":"485b1efc786da48b","type":"ui_gauge","z":"a7681
dcb045c877f","name":"","group":"7ae568990ac06036","order":3,"width":0,"height":0,"gtype":"gage","title"
:"Battery capacity
remaining","label":"%","format":"{{value}}","min":0,"max":"100","colors":["#ff0000","#ffff00","#00ff00"
],"seg1":"25","seg2":"70","className":"","x":640,"y":1140,"wires":[]},{"id":"f57882ebd6dc0c9a","type":"
debug","z":"a7681dcb045c877f","name":"Battery capacity
remaining","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targe
tType":"msg","statusVal":"","statusType":"auto","x":640,"y":1100,"wires":[]},{"id":"c833ed6f9c40f502","
type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/vehicle/battery/current","qos":"2","datatype":"
auto","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":270,"y":1260,"wires":[["9
9eaf589a2ebb9a4","5f5ddd0038f5e8e0"]]},{"id":"99eaf589a2ebb9a4","type":"debug","z":"a7681dcb045c877f","
name":"Battery current
draw","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType
":"msg","statusVal":"","statusType":"auto","x":620,"y":1220,"wires":[]},{"id":"4e59a118128c3fd0","type"
:"ui_gauge","z":"a7681dcb045c877f","name":"","group":"7ae568990ac06036","order":2,"width":0,"height":0,
"gtype":"gage","title":"Battery current
draw","label":"Ampere","format":"{{value}}","min":0,"max":"45","colors":["#00ff00","#ffff00","#ff0000"]
,"seg1":"25","seg2":"35","className":"","x":1100,"y":1260,"wires":[]},{"id":"5f5ddd0038f5e8e0","type":"
range","z":"a7681dcb045c877f","minin":"0","maxin":"45000","minout":"0","maxout":"45","action":"scale","
round":false,"property":"payload","name":"","x":570,"y":1260,"wires":[["354ae367a1dc88cc"]]},{"id":"db0
f5489a99a4b5e","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/vehicle/battery/voltage","qos":"2","datatype":"
auto","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":270,"y":1380,"wires":[["9
5d0100095cdc038","10a846f0e3452425"]]},{"id":"10a846f0e3452425","type":"debug","z":"a7681dcb045c877f","
name":"Battery
voltage","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetT
ype":"msg","statusVal":"","statusType":"auto","x":600,"y":1340,"wires":[]},{"id":"f53ec4a2c4e7a02c","ty
pe":"ui_gauge","z":"a7681dcb045c877f","name":"","group":"7ae568990ac06036","order":1,"width":0,"height"
:0,"gtype":"gage","title":"Battery
voltage","label":"Volt","format":"{{value}}","min":0,"max":"17","colors":["#ff0000","#ffff00","#00ff00"
```

```
],"seg1":"14","seg2":"15","className":"","x":1080,"y":1380,"wires":[]},{"id":"95d0100095cdc038","type":
"range","z":"a7681dcb045c877f","minin":"0","maxin":"17000","minout":"0","maxout":"17","action":"scale",
"round":false,"property":"payload","name":"","x":570,"y":1380,"wires":[["031b358cf8f4361d"]]},{"id":"3b
c5395cf33b0d67","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/mission/item_reached","qos":"2","datatype":"aut
o","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":270,"y":1740,"wires":[["9a9c
10b8eb5821a9","cc10a7097905c902"]]},{"id":"e6c7259855bf266b","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/communications/packetloss","qos":"2","datatype"
:"auto","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":260,"y":1500,"wires":[[
"9775d6b6120efd7d","d7cc9b2a2a9957f3"]]},{"id":"9775d6b6120efd7d","type":"debug","z":"a7681dcb045c877f"
,"name":"Communications packet
loss","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType
":"msg","statusVal":"","statusType":"auto","x":640,"y":1460,"wires":[]},{"id":"f2feb05909a0cbb1","type"
:"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/cpu/load","qos":"2","datatype":"auto","broker":
"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":320,"y":1620,"wires":[["96ee03d1c40d1767
","c80dfecba5c8fc45"]]},{"id":"96ee03d1c40d1767","type":"range","z":"a7681dcb045c877f","minin":"0","max
in":"1000","minout":"0","maxout":"100","action":"scale","round":false,"property":"payload","name":"","x
":580,"y":1620,"wires":[["6c4b759001edbca5"]]},{"id":"c80dfecba5c8fc45","type":"debug","z":"a7681dcb045
c877f","name":"Autopilot CPU
load","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType
":"msg","statusVal":"","statusType":"auto","x":610,"y":1580,"wires":[]},{"id":"9a9c10b8eb5821a9","type"
:"ui_text","z":"a7681dcb045c877f","group":"374b8b3839dbaf9e","order":4,"width":"6","height":"2","name":
"","label":"Mission status","format":"{{msg.payload}}","layout":"col-
center","className":"","x":600,"y":1740,"wires":[]},{"id":"cc10a7097905c902","type":"debug","z":"a7681d
cb045c877f","name":"Mission item
reached","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetT
ype":"msg","statusVal":"","statusType":"auto","x":620,"y":1700,"wires":[]},{"id":"b6b21af0e64fa143","ty
pe":"ui_gauge","z":"a7681dcb045c877f","name":"","group":"c245161a4a1c70c6","order":4,"width":0,"height"
:0,"gtype":"gage","title":"Autopilot CPU
load","label":"%","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"se
g1":"33","seg2":"66","className":"","x":1090,"y":1620,"wires":[]},{"id":"d7cc9b2a2a9957f3","type":"ui_g
auge","z":"a7681dcb045c877f","name":"","group":"c245161a4a1c70c6","order":5,"width":0,"height":0,"gtype
":"gage","title":"Communications packet
loss","label":"%","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"se
g1":"","seg2":"","className":"","x":640,"y":1500,"wires":[]},{"id":"630a02b5d2ea4bed","type":"change","
z":"a7681dcb045c877f","name":"","rules":[{"t":"change","p":"payload","pt":"msg","from":"0","fromt":"str
","to":"No GPS
connected","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"1","fromt":"str","to":"No
position information, GPS is
connected","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"2","fromt":"str","to":"2D
position","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"3","fromt":"str","to":"3D
position","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"4","fromt":"str","to":"DGPS/SBAS
aided 3D
position","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"5","fromt":"str","to":"RTK float,
3D position","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"6","fromt":"str","to":"RTK
Fixed, 3D
position","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"7","fromt":"str","to":"Static
fixed, typically used for base
```

```
stations","tot":"str"},{"t":"change","p":"payload","pt":"msg","from":"8","fromt":"str","to":"PPP, 3D
position.","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":600,"y":780,"wires
":[["f96c7962431083f8"]]},{"id":"6c4b759001edbca5","type":"function","z":"a7681dcb045c877f","name":"","
func":"var numb = msg.payload;\nnumb = numb.toFixed(1);\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":820,"y":1620,"wires":[["b6b21af
0e64fa143"]]},{"id":"031b358cf8f4361d","type":"function","z":"a7681dcb045c877f","name":"","func":"var
numb = msg.payload;\nnumb = numb.toFixed(3);\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":820,"y":1380,"wires":[["f53ec4a
2c4e7a02c"]]},{"id":"354ae367a1dc88cc","type":"function","z":"a7681dcb045c877f","name":"","func":"var
numb = msg.payload;\nnumb = numb*10;\nnumb = numb.toFixed(3);\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":820,"y":1260,"wires":[["4e59a11
8128c3fd0"]]},{"id":"cb9bdfa55c4332df","type":"debug","z":"a7681dcb045c877f","name":"Command
ACK","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":540,"y":180,"wires":[]},{"id":"4ef72a6e42f9721d","type":"
worldmap in","z":"a7681dcb045c877f","name":"Mission
Coordinates","path":"/worldmap/mission","events":"connect,disconnect,point,layer,bounds,files,draw,othe
r","x":330,"y":1880,"wires":[["c82410377d5225dc","143de68201d30392"]]},{"id":"d3be47b178a8216c","type":
"ui_worldmap","z":"a7681dcb045c877f","group":"cf03a5ffb3f477cd","order":1,"width":"6","height":"6","nam
e":"","lat":"38.34","lon":"23.67","zoom":"","layer":"OSMC","cluster":"","maxage":"","usermenu":"hide","
layers":"hide","panit":"false","panlock":"false","zoomlock":"false","hiderightclick":"false","coords":"
deg","showgrid":"false","allowFileDrop":"false","path":"/worldmap/mission","overlist":"DR,CO,RA,DN,SN,H
M","maplist":"OSMG,OSMC,EsriC,EsriS,EsriT,EsriDG,UKOS,SW","mapname":"","mapurl":"","mapopt":"","mapwms"
:false,"x":1570,"y":2020,"wires":[]},{"id":"aec572880c9d122b","type":"ui_worldmap","z":"a7681dcb045c877
f","group":"8afb3977543fc7ab","order":1,"width":"6","height":"6","name":"","lat":"38.34","lon":"23.67",
"zoom":"","layer":"OSMC","cluster":"","maxage":"","usermenu":"hide","layers":"hide","panit":"false","pa
nlock":"false","zoomlock":"false","hiderightclick":"false","coords":"deg","showgrid":"false","allowFile
Drop":"false","path":"/worldmap/RTL","overlist":"DR,CO,RA,DN,HM","maplist":"OSMG,OSMC,EsriC,EsriS,EsriT
,EsriDG,UKOS,SW","mapname":"","mapurl":"","mapopt":"","mapwms":false,"x":1620,"y":2360,"wires":[]},{"id
":"c82410377d5225dc","type":"split","z":"a7681dcb045c877f","name":"","splt":"\\n","spltType":"str","arr
aySplt":1,"arraySpltType":"len","stream":false,"addname":"topic","x":560,"y":1880,"wires":[["150858cdd4
18a034"]]},{"id":"150858cdd418a034","type":"switch","z":"a7681dcb045c877f","name":"","property":"topic"
,"propertyType":"msg","rules":[{"t":"eq","v":"lat","vt":"str"},{"t":"eq","v":"lon","vt":"str"}],"checka
ll":"true","repair":false,"outputs":2,"x":750,"y":1880,"wires":[["89d085fcccc3e67b"],["b6da040f06f1574b
"]]},{"id":"89d085fcccc3e67b","type":"function","z":"a7681dcb045c877f","name":"","func":"var numb =
msg.payload;\nnumb = numb.toFixed(7);\nnumb =  numb.split('.').join('')\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":900,"y":1840,"wires":[["e722948
8c21e7d63","dfc063cd502dbf52"]]},{"id":"b6da040f06f1574b","type":"function","z":"a7681dcb045c877f","nam
e":"","func":"var numb = msg.payload;\nnumb = numb.toFixed(7);\nnumb =
numb.split('.').join('')\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":900,"y":1920,"wires":[["e722948
8c21e7d63","dfc063cd502dbf52"]]},{"id":"e7229488c21e7d63","type":"join","z":"a7681dcb045c877f","name":"
","mode":"custom","build":"object","property":"payload","propertyType":"msg","key":"topic","joiner":"\\
n","joinerType":"str","accumulate":true,"timeout":"","count":"2","reduceRight":false,"reduceExp":"","re
duceInit":"","reduceInitType":"","reduceFixup":"","x":1150,"y":1920,"wires":[["9df01390fcb23c11"]]},{"i
d":"dfc063cd502dbf52","type":"join","z":"a7681dcb045c877f","name":"","mode":"custom","build":"string","
property":"payload","propertyType":"msg","key":"topic","joiner":",","joinerType":"str","accumulate":fal
se,"timeout":"","count":"2","reduceRight":false,"reduceExp":"","reduceInit":"","reduceInitType":"","red
uceFixup":"","x":1150,"y":1840,"wires":[[]]},{"id":"9fd55383e0df98e9","type":"worldmap
in","z":"a7681dcb045c877f","name":"RTL
```

```
Coordinates","path":"/worldmap/RTL","events":"connect,disconnect,point,layer,bounds,files,draw,other","
x":340,"y":2220,"wires":[["38731449b676f9b5"]]},{"id":"38731449b676f9b5","type":"split","z":"a7681dcb04
5c877f","name":"","splt":"\\n","spltType":"str","arraySplt":1,"arraySpltType":"len","stream":false,"add
name":"topic","x":560,"y":2220,"wires":[["31e2a7d81e9e8c52"]]},{"id":"31e2a7d81e9e8c52","type":"switch"
,"z":"a7681dcb045c877f","name":"","property":"topic","propertyType":"msg","rules":[{"t":"eq","v":"lat",
"vt":"str"},{"t":"eq","v":"lon","vt":"str"}],"checkall":"true","repair":false,"outputs":2,"x":750,"y":2
220,"wires":[["b3dd9d877811ea33"],["f618fef118fcc98f"]]},{"id":"b3dd9d877811ea33","type":"function","z"
:"a7681dcb045c877f","name":"","func":"var numb = msg.payload;\nnumb = numb.toFixed(7);\nnumb =
numb.split('.').join('')\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":900,"y":2180,"wires":[["b6b4428
d9ae97287","41af116d122d35a6"]]},{"id":"f618fef118fcc98f","type":"function","z":"a7681dcb045c877f","nam
e":"","func":"var numb = msg.payload;\nnumb = numb.toFixed(7);\nnumb =
numb.split('.').join('')\nmsg.payload=numb\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":900,"y":2260,"wires":[["b6b4428
d9ae97287","41af116d122d35a6"]]},{"id":"b6b4428d9ae97287","type":"join","z":"a7681dcb045c877f","name":"
","mode":"custom","build":"object","property":"payload","propertyType":"msg","key":"topic","joiner":"\\
n","joinerType":"str","accumulate":true,"timeout":"","count":"2","reduceRight":false,"reduceExp":"","re
duceInit":"","reduceInitType":"","reduceFixup":"","x":1150,"y":2260,"wires":[["99a53d86587e4d1d"]]},{"i
d":"41af116d122d35a6","type":"join","z":"a7681dcb045c877f","name":"","mode":"custom","build":"string","
property":"payload","propertyType":"msg","key":"topic","joiner":",","joinerType":"str","accumulate":fal
se,"timeout":"","count":"2","reduceRight":false,"reduceExp":"","reduceInit":"","reduceInitType":"","red
uceFixup":"","x":1150,"y":2180,"wires":[[]]},{"id":"e0b35f28d3b717a7","type":"comment","z":"a7681dcb045
c877f","name":"Sends coordinates to autopilots specific
topic","info":"","x":1290,"y":2320,"wires":[]},{"id":"8baf5ad89793b704","type":"comment","z":"a7681dcb0
45c877f","name":"Sets the missions
coordinates","info":"","x":340,"y":2400,"wires":[]},{"id":"fdf07b86e59520f4","type":"debug","z":"a7681d
cb045c877f","name":"Final
wordlmap","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","target
Type":"msg","statusVal":"","statusType":"auto","x":1820,"y":940,"wires":[]},{"id":"d47ecec1cb12ac9c","t
ype":"function","z":"a7681dcb045c877f","name":"","func":"var lat = msg.payload.lat;\nvar lon =
msg.payload.lon;\nmsg.payload = {};\nmsg.payload.name = \"test\";\nmsg.payload.lat =
flow.payload.lat;\nmsg.payload.lon = flow.payload.lon;\n//msg.payload.lat =
51.645294049305406;\n//msg.payload.lon = 7.756347656250001;\nmsg.payload.icon =
\"circle\";\nmsg.payload.iconColor = \"#910000\";\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1580,"y":940,"wires":[["fdf07b8
6e59520f4","becfacef792a749d"]]},{"id":"becfacef792a749d","type":"worldmap","z":"a7681dcb045c877f","nam
e":"","lat":"38.34","lon":"23.67","zoom":"16","layer":"OSMG","cluster":"","maxage":"","usermenu":"hide"
,"layers":"hide","panit":"false","panlock":"false","zoomlock":"false","hiderightclick":"false","coords"
:"none","showgrid":"false","path":"/worldmap/current","overlist":"DR,CO,RA,DN,HM","maplist":"OSMG,OSMC,
EsriC,EsriS,EsriT,EsriO,EsriDG,NatGeo,UKOS,OpTop,SW","mapname":"","mapurl":"","mapopt":"","mapwms":fals
e,"x":1830,"y":880,"wires":[]},{"id":"7d8551fc9a8a3b1b","type":"ui_worldmap","z":"a7681dcb045c877f","gr
oup":"374b8b3839dbaf9e","order":1,"width":"6","height":"6","name":"","lat":"38.34","lon":"23.67","zoom"
:"","layer":"OSMC","cluster":"","maxage":"","usermenu":"hide","layers":"hide","panit":"false","panlock"
:"false","zoomlock":"false","hiderightclick":"true","coords":"deg","showgrid":"false","allowFileDrop":"
false","path":"/worldmap/current","overlist":"DR,CO,RA,DN,SN,HM","maplist":"OSMG,OSMC,EsriC,EsriS,EsriT
,EsriDG,UKOS,SW","mapname":"","mapurl":"","mapopt":"","mapwms":false,"x":1910,"y":720,"wires":[]},{"id"
:"8c7420d3f43b10c2","type":"inject","z":"a7681dcb045c877f","name":"","props":[{"p":"payload"},{"p":"top
ic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payloadType":"str","
x":1370,"y":940,"wires":[["d47ecec1cb12ac9c"]]},{"id":"143de68201d30392","type":"debug","z":"a7681dcb04
```

```
5c877f","name":"worldmap
node","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType
":"msg","statusVal":"","statusType":"auto","x":320,"y":1820,"wires":[]},{"id":"ceb3cf6f7c3ea775","type"
:"function","z":"a7681dcb045c877f","name":"","func":"if (msg.payload.action == \"point\"){\n   return
msg;
\n}\n","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":960,"y":1780,"wires":[["b202fb
546a86aee1","68d249909e61a5a6"]]},{"id":"b202fb546a86aee1","type":"debug","z":"a7681dcb045c877f","name"
:"filter
msg","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":980,"y":1720,"wires":[]},{"id":"68d249909e61a5a6","type":
"function","z":"a7681dcb045c877f","name":"","func":"\ndelete msg.payload.layer\ndelete
msg.topic\ndelete msg.payload.hdg\ndelete msg.payload.ttl\ndelete msg.payload.action\ndelete
msg.payload.draggable\nmsg.payload.name = \"frommap\";\nmsg.payload.icon =
\"circle\";\nmsg.payload.iconColor = \"#910000\";\n\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1180,"y":1780,"wires":[["8e8597
0d5d3665ac"]]},{"id":"8e85970d5d3665ac","type":"debug","z":"a7681dcb045c877f","name":"","active":false,
"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto"
,"x":1410,"y":1780,"wires":[]},{"id":"9396df79f8fce068","type":"function","z":"a7681dcb045c877f","name"
:"","func":"msg.payload.name =\"current_gps\";\nmsg.payload.lat =
msg.payload.latitude/10000000;\nmsg.payload.lon = msg.payload.longitude/10000000;\ndelete
msg.payload.latitude;\ndelete msg.payload.longitude;\nmsg.payload.icon
=\"circle\";\nmsg.payload.iconColor =\"#300000\";\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1220,"y":820,"wires":[["789d152
4fc54c448","becfacef792a749d"]]},{"id":"789d1524fc54c448","type":"debug","z":"a7681dcb045c877f","name":
"next
step","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":1800,"y":820,"wires":[]},{"id":"93701550747b010f","type":
"function","z":"a7681dcb045c877f","name":"","func":"msg.payload.name =\"rtl_gps\";\nmsg.payload.lat =
msg.payload.latitude/10000000;\nmsg.payload.lon = msg.payload.longitude/10000000;\ndelete
msg.payload.latitude;\ndelete msg.payload.longitude;\nmsg.payload.icon
=\"circle\";\nmsg.payload.iconColor =\"#300000\";\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1220,"y":560,"wires":[["2107c52
5b4458a86","898485fc086abadd"]]},{"id":"2107c525b4458a86","type":"debug","z":"a7681dcb045c877f","name":
"next
step","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType"
:"msg","statusVal":"","statusType":"auto","x":1800,"y":560,"wires":[]},{"id":"898485fc086abadd","type":
"worldmap","z":"a7681dcb045c877f","name":"","lat":"38.34","lon":"23.67","zoom":"16","layer":"OSMC","clu
ster":"","maxage":"","usermenu":"hide","layers":"hide","panit":"false","panlock":"false","zoomlock":"fa
lse","hiderightclick":"false","coords":"deg","showgrid":"false","path":"/worldmap/RTL","overlist":"DR,C
O,RA,DN,HM","maplist":"OSMG,OSMC,EsriC,EsriS,EsriT,EsriO,EsriDG,NatGeo,UKOS,OpTop,SW","mapname":"","map
url":"","mapopt":"","mapwms":false,"x":1820,"y":600,"wires":[]},{"id":"8e64aa90e7bc7c2a","type":"mqtt
in","z":"a7681dcb045c877f","name":"","topic":"autopilot/companion_computer/RSSI","qos":"2","datatype":"
auto","broker":"e8efa29ec5eec79c","nl":false,"rap":true,"rh":0,"inputs":0,"x":260,"y":660,"wires":[["62
69d4c1bc4ed3af"]]},{"id":"6269d4c1bc4ed3af","type":"ui_gauge","z":"a7681dcb045c877f","name":"","group":
"c245161a4a1c70c6","order":9,"width":0,"height":0,"gtype":"gage","title":"Companion Computer
RSSI","label":"dbm","format":"{{value}}","min":"-
100","max":"0","colors":["#ff0000","#ffff00","#00ff00"],"seg1":"-85","seg2":"-
55","className":"","x":640,"y":660,"wires":[]},{"id":"e8efa29ec5eec79c","type":"mqtt-
broker","name":"Boat
```

mqtt","broker":"83.212.77.23","port":"1883","clientid":"","autoConnect":true,"usetls":false,"protocolVe
rsion":"4","keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","birth
Msg":{},"closeTopic":"","closeQos":"0","closePayload":"","closeMsg":{},"willTopic":"","willQos":"0","wi
llPayload":"","willMsg":{},"sessionExpiry":""},{"id":"0f1f8234940685b4","type":"ui_group","name":"Debug
Data","tab":"ff25ab1d26c005a4","order":6,"disp":true,"width":"6","collapse":false,"className":""},{"id"
:"cf03a5ffb3f477cd","type":"ui_group","name":"Mission Data &
Control","tab":"ff25ab1d26c005a4","order":1,"disp":true,"width":"6","collapse":false,"className":""},{"
id":"8afb3977543fc7ab","type":"ui_group","name":"RTL GPS Data &
Control","tab":"ff25ab1d26c005a4","order":3,"disp":true,"width":"6","collapse":false,"className":""},{"
id":"c245161a4a1c70c6","type":"ui_group","name":"Autopilot Data &
Control","tab":"ff25ab1d26c005a4","order":4,"disp":true,"width":"6","collapse":false,"className":""},{"
id":"374b8b3839dbaf9e","type":"ui_group","name":"Current Mission Data &
Control","tab":"ff25ab1d26c005a4","order":2,"disp":true,"width":"6","collapse":false,"className":""},{"
id":"7ae568990ac06036","type":"ui_group","name":"Battery
Data","tab":"ff25ab1d26c005a4","order":5,"disp":true,"width":"6","collapse":false,"className":""},{"id"
:"ff25ab1d26c005a4","type":"ui_tab","name":"Boat Control
Panel","icon":"dashboard","disabled":false,"hidden":false}]