UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

PROJECT OF ATTITUDE CONTROL SYSTEM FOR 1U CUBESAT

FINAL OUTPUT REPORT

Student: Daniel Castillo Bonillo

Directors: Dr. David Gonzalez Diez, Dr. Jordi Voltas Aguilar

Double degree in Mechanical Engineering and Electronic, Industrial and Automatic Engineering

Superior School of Industrial, Aerospace and Audiovisual Engineering of Terrassa
Technical University of Catalonia — BarcelonaTech

September 27, 2022

Abstract

The high costs involved in carrying out any real test related to nanosatellite research, as well as the large amount of human and temporary resources that these requires, make it impossible for any educational institution to offer its students the possibility to be formed in this area. This fact has generated a tendency for these educational institutions to replace the practical teaching on this field by a more theoretical approach only practiced through computational simulations.

However, this project carries out the development, design, and manufacture, at both mechanical and electronic level, of a test prototype focused on the study of the attitude control systems of a U1 CubeSat, as well as the new typologies of photovoltaic panels based on Origami patterns. This prototype includes both the design of an unfolding mechanism for these types of deployable photovoltaic panels, and the development and implementation of the attitude control device in which the mechanism is incorporated. Moreover, this project also covers notable improvements of both the measurement subsystems and the position control algorithm.

The proposed design of the U1 CubeSat prototype frame, the attitude control systems and the photovoltaic panel deployment mechanism is modular. This implies that each of these subsystems can be operated independently from the rest, which greatly eases the study of each one of them and fits perfectly with the ideology of modular CubeSats, which can improve their functionalities depending on the added elements. Considering so, this prototype incorporates an attitude control device with a single reaction wheel and its corresponding MPU position sensor, as well as a mechanism for deploying photovoltaic panels based on Origami patterns. It should be noted that each of these systems includes its corresponding power supply, all controlled by an STM32 BluePill microcontroller and communicated through an HC-06 Bluetooth module.

To facilitate and speed up the prototyping process, all the parts defined throughout the project have been manufactured using FDM additive manufacturing technologies, and modeled using SolidWorks mechanical CAD software, which results in a significant reduction in development and manufacturing costs.

Keywords:
CubeSat, Attitude control system, Photovoltaic panel, Origami, Reaction wheel, Prototype, MPU

Resumen

Los altos costes que implica la ejecución de cualquier ensayo real relacionado con la investigación en nanosatélites, así como la gran cantidad que recursos tanto humanos como temporales que estos requieren imposibilitan que cualquier institución educativa pueda ofrecer a su alumnado de forma generalizada la posibilidad de formarse en este ámbito. Este hecho ha generado una tendencia por parte de estas instituciones educativas superiores a sustituir el enfoque de enseñanza práctica por uno más orientado a al estudio teórico o la práctica a través de simulaciones computacionales.

En este proyecto se lleva a cabo el desarrollo, diseño y fabricación a nivel mecánico y electrónico de un prototipo de pruebas enfocado al estudio tanto de los sistemas de control de actitud de un U1 CubeSat como de nuevas tipologías de placas fotovoltaicas basadas en patrones de Origami. En este prototipo se incluye tanto el diseño de un mecanismo de despliegue de dichas tipologías de placas fotovoltaicas desplegables como el desarrollo e implementación del dispositivo de control de actitud en el cual se incorpora dicho mecanismo. El desarrollo de este dispositivo también engloba una notable mejora en los subsistemas de medición, así como en el algoritmo de control de posición.

El diseño tanto del bastidor del prototipo del U1 CubeSat como de los sistemas de control de actitud y el mecanismo de despliegue de placas fotovoltaicas es modular. Esto implica que cada uno de estos subsistemas puede funcionar de forma independiente al resto, lo que facilita en gran medida el estudio independiente de cada uno ellos y encaja a la perfección con la ideología de los CubeSats modulares, los cuales pueden variar sus funcionalidades dependiendo de los elementos añadidos.
Este prototipo incluye un dispositivo de control de actitud con 1 volante de reacción y su correspondiente sensor de posición MPU además de un mecanismo de despliegue de placas fotovoltaicas basadas en patrones de Origami, cabe comentar que cada uno de estos sistemas incluye su correspondiente fuente de alimentación independiente, todo ello controlado por un microcontrolador STM32 BluePill y comunicado a través de un módulo Bluetooth HC-06.

Para facilitar y agilizar el proceso de prototipado, todas las piezas definidas a lo largo del proyecto han sido fabricadas mediante tecnologías de fabricación aditiva FDM y modeladas mediante el software de CAD mecánico SolidWorks, lo que resulta en una significativa reducción de los costes de desarrollo y fabricación.

Palabras clave:
CubeSat, Control de Actitud, Placa fotovoltaica, Origami, Rueda de reacción, Prototipo, MPU

Acknowledgments

The author wishes to express his deepest gratitude to his supervisor Prof. Dr. David Gonzálvez and Prof. Dr. Jordi Voltas for their advice, interest, and encouragement even in times of great pressure and stress for both parts. I could not have imagined accomplishing what I have achieved without our many hours of weekly meetings, and your guidance so that, in some moments, I could put my feet on the ground and focus on the goal, leaving aside my obsessions. Your advice has been essential to the success of this project.

I would also like to specially thank the TIEG research group members Prof. Dr. Javier Gago, Prof. Dr. Manel Lamich, and Prof. Dr. Antoni Arias for their encouragement and interest in my advances and research, although they were not formally obliged to do so.

A big thanks to my colleagues Yi Qiang Ji Zhang and Adrià Pérez without whose outstanding previous work I would never have been able to complete such a deep research and development of this project.

I would also like to include Joan Serrano and Ivan Gonzálvez in this acknowledgment, since thanks to their work I have achieved a deeper comprehension of ADCS nanosatellites subsystems.

Thanks to all the professors and colleagues that I have met during these 5 years, for their advice, comfort, help, and both academic and personal teachings, which have made me become a better person.

Finally, I would like to dedicate this work to my family: my father Jaime Castillo, my mother Josefina Bonillo, and my brother Álvaro Castillo for supporting me with this endeavor and always cheering me up to improve myself every day. I will be always indebted to them and I feel blessed and privileged to have them as my family.

Declaration on Honour

I declare that,

the work in this Degree Thesis is completely my own work,

no part of this Degree Thesis is taken from other people's work without giving them credit,

all references have been clearly cited,

I'm authorised to make use of the research group TIEG and DISEN related information I'm providing in this document.

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by The Technical University of Catalonia - BarcelonaTECH.

Daniel Castillo Bonillo		27 September 2022
Name of the student	Signature	Date

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ADCS	Attitude Determination and Control System
CCW	Counter Clock Wise
CW	Clock Wise
DC	Direct Current
DMP	Digital Motion Processor
DoF	Degree of Freedom
EPS	Electric Power System
FDM	Fused Deposition Modeling
FOS	Factor of Safety
GEO	Geostationary Orbit
HP MJF	HP Multi Jet Fusion
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LEO	Low Earth Orbit
OBC	On Board Computer
PCB	Printed Circuit Board
PID	Proportional Integral Derivative
PLATHON	Integrated simulation PLATform of Optical communications in Nanosatellites
PWM	Pulse Width Modulation
RW	Reaction Wheel
SCK/SCL	Serial Clock Line
SDA	Serial Data Line
SPI	Serial Peripheral Interface
ST	Setting Time
1U	One unit

Algebraic symbols

A	Area
a	apothem
α	angle (Except in equation 4.2 where is angular acceleration)
d	diameter
e	error signal
F	Force
G	Transfer function
h	height
I	Momentum of inertia (Except in Table 4.6 where is defined as J)
K	Control constant
L	Dodecagon edge longitude
l	longitude
M	Momentum
m	mass
n	number (quantity)
$\dot{q}$"	area related solar flux
r	radium
u	control signal
ξ	Damping coefficient
ϑ	angular position
η	efficiency
ω	angular speed

Chapter 1

Introduction

1.1 Objectives

First of all, it must be mentioned that this project is based on the previous bachelor's thesis of Yi Qiang Ji Zhang [1] and Adrià Pérez Fernández [2], and it is focused on the improvement of their work on the development of CubeSats subsystems.
The main goal pursued during the execution of this project is divided into two different parts, which will give a specific solution belonging to a different technical field. Therefore, and despite all of them being related to CubeSats technologies, the independent technical field required for this project are the following:

- **Mechanical field:** The most important objective in this part of the project will be the design and manufacturing of a functional prototype of a solar panel deployment system based on origami architectures. Some of the aspects that will be studied during this fragment of the final report will be the power requirements of the complete system, in order to do a proper sizing of the whole system, and the quantification and documentation of the mechanical specifications of the proposed final solution in terms of mechanical and functional tests.

- **Electronic field:** The main and most relevant CubeSat subsystem that is wanted to be developed in this fragment of the project is the attitude control subsystem, which is in charge of the orientation of the spacecraft once it has been deployed. Some of the most significant results to be given in this part are the programming code used to perform the maneuvers and a detailed explanation of the used hardware.

Finally, once developed the subsystems already mentioned in the two previous paragraphs, the combination of both solutions will give a complete and multidisciplinary answer to the desired improvement of the test CubSat prototype. Later, the final report will be completed with performance tests that proof the viability of the proposed solution.

1.2 Justification

CubeSats appeared as a teaching tool in universities during the '90s but immediately shifted from university labs to commercial applications and business. Nowadays, they have become a mature standard in the aerospace industry, and the first CubeSat-based constellations have appeared to offer global communication coverage.
Following this trend, the PLATHON (PLATform of Optical communications in Nanosatellites) Research Group aims to ease the prototyping and testing process of CubeSat technologies by developing a HiTL (Hardware-in-the-Loop) platform, which once integrated with a network and orbital simulator will allow the verification of the communication network of a nanosatellite cluster considering the operational constraints of the nodes it is constituted from [3]. This HiTL platform will have the main objective to give birth to a simulation method that will help in the improvement and, to some extent, minimization of some technical shortcomings in the following two areas related to such constellation-based missions:

- **Limitation on simulation tools:** Current communication network simulators used by the scientific community (NS2, SNS3, and Opensand) do not have the features required to design communications among neither: two nodes in fixed LEO (Low Earth Orbit), LEO and GEO (Geostationary Orbit) nodes, nor ground stations to orbit. Additionally, the power availability of nodes to establish proper communications are also not considered.
- **Lack of reliability:** CubeSat missions still have a large fail rate (55% in CubeSat from Universities and 33% from Companies), which is nowadays mainly derived from unexpected malfunctions on the Electric Power Systems (EPS) that occur in close to 40% of the failed deployment.

As a consequence, the following capabilities are studied, developed, and documented during this final degree thesis in order to solve the previously exposed limitations:

- Aiming to the improvement of some aspects of the first mentioned weakness of CubeSat HiTL platforms, it must be considered and included in the simulation process that one of the most important features that a CubeSat has to offer is the ability to communicate with other devices that surround it, either with ground stations or other satellites.
  Considering so, one of the core systems affecting the communication between devices is the attitude control subsystem, which is in charge of the spacecraft pointing and orientation accuracy. Therefore, its improvement will allow better control over the line-of-sight of optical communication devices and enhance communication between all in-orbit devices.

- Related to some of the already mentioned problems with Electric Power Systems (EPS), an origami-based deployable solar panel will be developed to allow and increase of the solar radiation incision surface of the CubeSat. This modification, aimed to increase the energy production of the EPS, is expected to increase the versatility of the usable systems in this kind of NanoSatellites.

1.3 Scope of the project

The final delivery of this project will contain the following documents, to ensure a proper justification and documentation of the proposed prototype improvements, as well as a detailed explanation of the final solution.

- **Final written report:** This document will contain the whole development, justification, and documentation of mechanical and electronic systems studied in this project. Moreover, the documentation of this report will also include a justification of the selected hardware, a detailed explanation of the steps followed to fulfill the required specifications, and an informed conclusion.

- **Blueprints of the developed pieces and assemblies:** Every single designed piece must be justified and represented with its respective dimensions and mechanical characteristics such as material, treatment, and surface quality.

- **Arduino codes of the attitude control system:** In order to provide a proper justification of the given software solution, a detailed explanation and exemplification of the attitude control code implemented in the spacecraft must be delivered.

- **Electrical diagrams of the wired connections:** Is of paramount importance to graphically represent every single connection performed in the hardware system to ensure the correct functionality of the final prototype. This representation can also simplify the understanding and possible replication of the circuit.

- **Datasheets of the used hardware:** To deliver all the required information for a proper justification of the proposed solution, the real datasheets of the used hardware are needed. This documentation will include the technical specifications used to develop the final given results.

- **Final developed prototype:** Since the main objective of this project is to design, justify, and manufacture a functional CubeSat prototype for testing purposes, the delivery of this physical device will be required. In addition, all the already mentioned documents are subjected to the delivery of this final prototype, which will be of great importance for the proper comprehension of the achieved goals.

1.4 Project requirements

The following list enumerates the basic requirements that the proposed solution has to respect to be considered as an optimum one.

- **Solar panels deployment requirements:** When talking about the specifications that a mechanical system composed of a great amount of relatively small pieces, is of paramount importance to predict and take into account how the whole device will react to the movement of these pieces. Consequently, the design process of this subsystem requires a great degree of dimensional precision.

- **Power requirements of the system:** To perform a proper dimensional design of the solar panel deployment system, is important to have into consideration the power requirements of the completed satellite prototype, to optimize the compactness while supplying enough power to allow the correct behavior of the whole device.

- **Material properties requirements:** The final proposed mechanical solution has to be developed to be manufactured with materials that can withstand extreme conditions, going from the exposition to great forces during the launching to very low pressure in spatial conditions.

- **Manufacturing method requirements:** According to the time and economical limitations, the manufacturing method used to develop the final CubeSat prototype must be reliable and accessible so as to speed up the whole prototyping process. As will be justified in Section 3.2.1.2, the finally selected addition manufacturing technology will be FDM 3D printing.

- **Attitude control precision required:** Due to the stated impact on communication capabilities, the pointing precision of the developed attitude control subsystem has an important role in the final behavior of the CubeSat after the device deployment. Because of this fact, this subsystem will need to fulfill demanding precision standards.

- **The programming language used:** Because of hardware requirements, all the control codes programmed during the development of this project will be codified in Arduino programming language.

- **Application requirements:** Because of the specific conditions to which the CubeSat spacecraft will be exposed, the design of both the mechanical and electrical subsystems will be studied to apply to these situations.

Chapter 2

State of the art

2.1 CubeSat

CubeSats are a standardization of nanosatellites (which present a weight range between 1-10kg) whose dimensions can not exceed 10x10x10cm in volume (when talking about 1U CubeSat). CubeSat specifications were born in California Polytechnic State University [4] and Stanford University as a consequence of the need for cheap technologies in order for the universities to perform space investigation projects in LEO orbits (Low Earth Orbits).

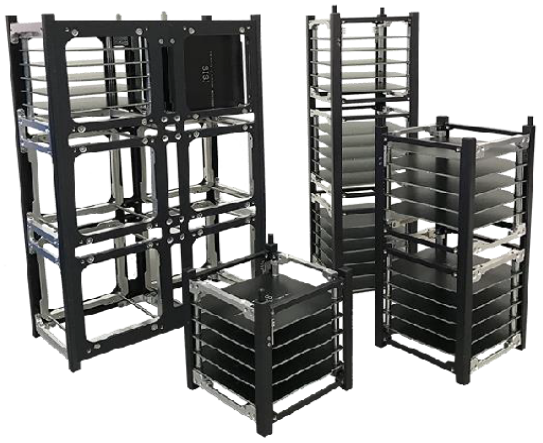


Figure 2.1 CubeSat structure 1U and formation of multiple units up to 6U. Source: [5]

The main advantage of CubeSat technologies is, because of their simplicity, the great reduction in terms of design complexity, which allows the use of CubeSats in short-term projects presenting at the same time high efficiency. Due to its reduced dimensions, of 10x10x10cm in the case of 1U CubeSats, they do not imply a great charge when launching. Moreover, satellites of higher complexity can be also assembled by combination of several units, using units of 1U or 1/2U size to reach an spacecraft of up to 24U.

In the following image from the Canadian Space Agency (Figure 2.2), it can be seen an illustrative concept of the existing satellite typologies and their differentiation in terms of relative dimension towards the others:

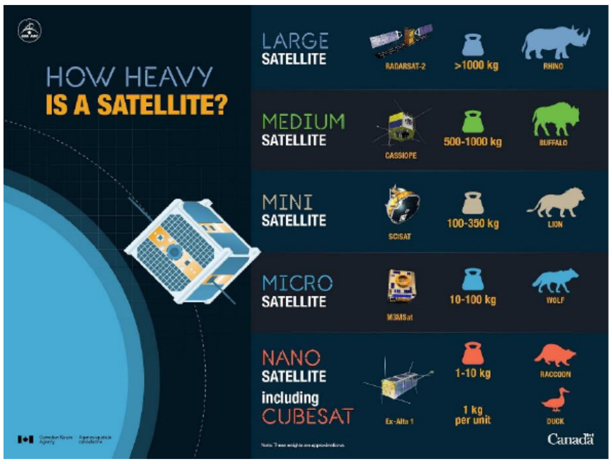


Figure 2.2 How heavy is a satellite? Source: [6]

Due to all these factors, a significant reduction in terms of launching cost is also taken into account when talking about CubeSat technologies, since they range between 50.000$ and 100.000$ depending on its complexity, and there is a continuous cost reduction derived from the emergence of new technologies.

Referring to the deployment process, due to the standardization of these technologies, it is carried out by using the named Poly-PicoSatellite Orbital Deployer (P-POD). The operation of this device is fairly simple: When a signal from the vehicle is received, the pre-mounted P-POD mechanisms (Figure 2.3) are deployed. After the deployment of the spacecraft, some control maneuvers are performed.

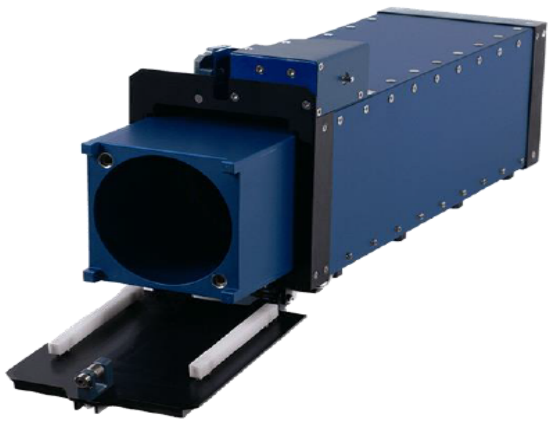


Figure 2.3 Poly-PicoSatellite Orbital Deployer (P-Pod). Source: [7]

2.2 Deployable Structures

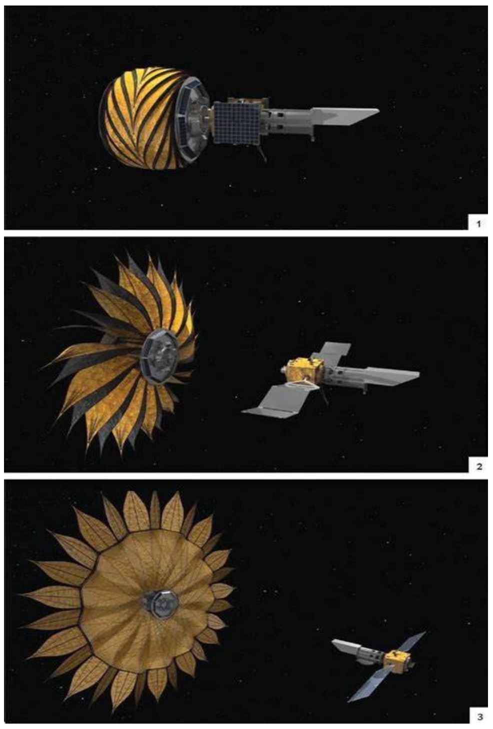
Since the mechanical part of this project is mainly focused on the development of a deployment system for Origami-based solar panel subsystems applied to nanosatellite devices, prior research of current deploying technologies used in aerospace applications can be of paramount importance for a proper understanding of the later on explained work. Considering so, some examples of the state of the art of this type of technology are presented below.

2.2.1 Starshade

Nowadays, astronomers and other scientists have been greatly limited in the investigation of Earth-sized exoplanets because of the methods that they are able to use, as they can only use indirect methods such as detecting the changes in starlight as a planet passes in front of its star, so as to give them some other observation method starshade was created. The starshade (also known as an external occulter) is a spacecraft that will enable telescopes in space to take pictures of planets orbiting faraway stars by blocking the glare of the star's light. The operating principle of this idea, is for the starshade to fly in front of the observation telescope and block the glare of the star's light before it enters the telescope, allowing the planet's reflected light to pass through and be collected.

As a consequence, starshade is designed to fly in formation up to 50,000 kilometers away from the telescope and deploy its origami structure in front of the star which is obstructing the observation of the studied exoplanet. Subsequently, the deployed flower-like starshade creates a shadow that allows the telescope to properly detect the light coming from the planets in the star's habitable zone, greatly enhancing the possibilities of exoplanetary research (Figure 2.4a).

When talking about real advances in starshade-based spacecraft, and in order to give a deeper framework to these technologies, it must be mentioned half-scale models of a 34m diameter deployable starshade structure were tested by NASA in summer of 2015. However, a successful deployment requires for 7m long petals to unfurl to form a shape with a precision of few human hair widths. After that, the origami pattern structure must then rotate, deploy and position the petals around a 20m diameter central disc structure with tolerance of fraction of a millimeter. Finally, this complex unfolded structure must then remain in this specific position for years in an extreme space environment with exceptionally high or low temperatures. Considering so, 20 deployments tests showed that the placement of the petals was consistently about a tenth of a millimeter from the correct position. Because of this successful test, it can be concluded that these structures for spacecraft applications can be implemented guaranteeing to meet their goals in every proposed functionality.



(a) **Figure 2.4** Starshade deployment process. Source: [8]

2.2.2 Deployable antennas

Some of the most commonly used deployable structures for aerospace applications, more specifically in nanosatellite technological research, are related to telecommunication and radiation sensing functionalities. This is because of these technologies requirement to have great precision in terms of deployed surface area and surface quality specifications.

In the following paragraphs, some of the most closely related deployable antenna structures to the deployable system proposed in this project will be briefly explained.

2.2.2.1 Rigid reflector antenna

This typology of a deployable antenna is based on a central hub and a number of rigid curved plates working as reflector surfaces [9][10]. The following image exemplifies this structure by using a physical model of the solid reflector deployable antenna with a diameter of 4.7 meters and a precision of 0.2mm (30GHz). This example of a rigid reflector antenna was made by a company named Dornier and ESA in 1987 (Figure 2.5).

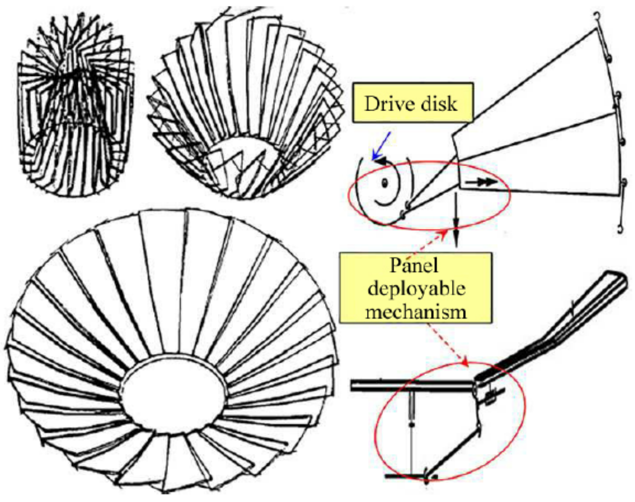


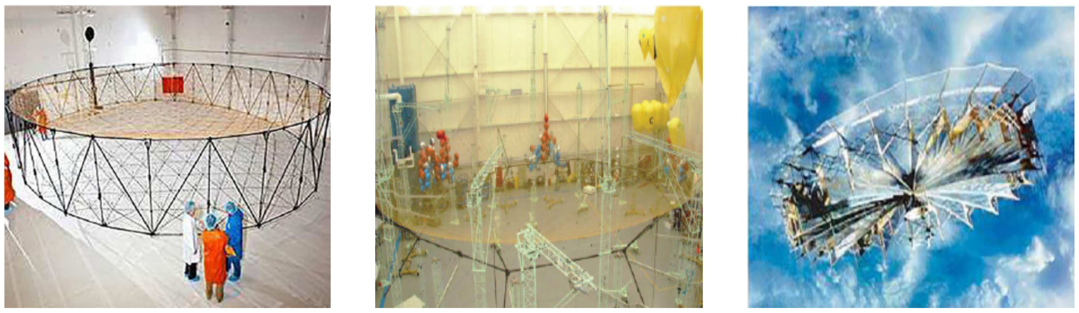
Figure 2.5 Petal type reflector antenna. Source: [11]

To further specify the manufacturing characteristics of this deployable antenna topology, as well as its functional properties, it must be mentioned that solid surface deployable antennas have been manufactured using a metal plate or Carbon fiber reinforced plastic (CFRP) coated with a metal reflective surface. Some other characteristic of this type of antenna is the advantage of the high precision of reflective surface that it offers, which not only can reach up to 0.13mm but also allows to be processed into a more ideal paraboloid if it displays a diameter of 10 meters [12].

Regarding the high accuracy of the profile after the unfolding of the antenna, this typology would be a good choice in the field of microwave remote sensing, such as microwave radiometers and scatterometers, and has been applied in several spacecraft [13]. It must also be mentioned that, because of its heavy structure, high cost, and large folding volume, it is rarely applied in large-diameter satellite antennas, which, at the same time, favors the possible application of these antenna typologies in nanosatellite design.

2.2.2.2 Mesh reflector loop antennas

Mesh reflector loop antennas refer to a whole typology of geometrically different deployable antennas composed of supporting unfolded structures composed by truss and cable nets. The truss can be placed in the periphery, center, or radial direction. Depending on the displacement of the truss, these antennas can be divided into a peripheral truss type, a Harris ring, an EGS loop antenna, and the like.



(a) Thuraya mesh antenna **(b)** SkyTerra Harris loop antenna **(c)** EGS loop antenna

Figure 2.6 Ring mesh reflector antenna. Source: [11]

- **Perimeter truss antenna:** Consists in multiple parallelogram units forming a closed loop structure that expands upon deployment to release a central mesh structure. Some examples of this structure are used in Thuraya, MBSat, and SMAP satellites, manufactured in the United States (Figure 2.6a).
- **Harris loop antenna:**This geometry consists of an inner deployable ring with regular polygon shape, which uses both upper and lower support diagonal rods to unfold a mesh attached to them. Examples of this typology are the 22m aperture antenna of US SkyTerra and MSV satellite (Figure 2.6b).
- **EGS loop antenna:** It is formed by two identical scissor-type expansion mechanisms that are attached to the mesh and spread it out upon opening. Russia and ESA jointly developed this type of antenna with a diameter of 12 meters (Figure 2.6c).

Considering these mesh-based antenna structures, it must be mentioned that the mesh-like nature of these typologies can be ideal for deployable mechanisms applied to nanosatellite technologies because of the versatility of shapes that a mesh can adopt without compromising the integrity of the material and the functionality of the whole device.

2.2.3 Roll Out Solar Array (ROSA)

Roll Out Solar Array (ROSA) and its larger version ISS Roll Out Solar Array (iROSA) are a typology of lightweight and flexible solar panels designed by NASA for spacecraft applications.

Since the space-bound payload has strict limitations in terms of mass and volume, ROSA technology was designed to overcome the bulky traditional panels with heavy hinge-based mechanical deployment mechanisms, which are still usually applied to power generation in satellite applications but show not to be the optimal solution [14]. Consequently, this new solar panel technology is shown to outperform the traditional systems in terms of power to weight ratio, since it is able to reach a much higher power generation while keeping much lower weight, allowing to overcome strict payload limitations. This new systems show to be 20% lighter (with a total mass of about 325 kg)[15] and four times more compact in volume than an equal performance rigid panel.

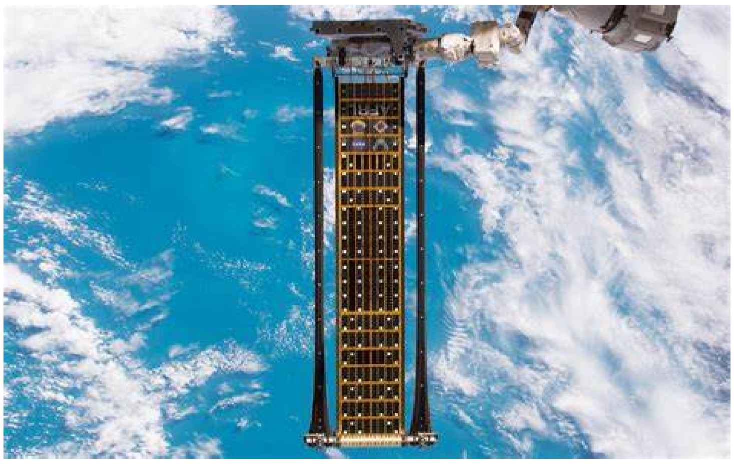


Figure 2.7 Roll Out Solar Array (ROSA) structure. Source: [15]

The function principle of ROSA flexible solar panels (Figure 2.7) is similar to the concept of a measuring tape, which rolls over itself in order to reduce its volume. Given so, this new solar array design rolls to form a compact cylinder during the launching process, displaying significantly lower volume and mass performances, which implies notable cost reductions in the whole process while enhancing the potential power generation performance of the spacecraft. ROSA has a central wing made of a flexible material whose objective is to support the photovoltaic cells in charge of power generation. Both sides of the wing have two narrow arms which are arranged longitudinally all over the matrix to give support to it. These narrow arms are flat tubes made of a rigid composite material, rolled up lengthwise.

Some other important property of this typology is the unfolding method, which does not require any motor to unfold. The deployment of this structure is achieved by using potential energy previously stored in the lateral arms by coiling them and producing angular tension, which can unfold the whole structure when transitions from a coil shape to a straight support arm is allowed. By using the same principle, the solar wings are then deployed due to strain energy in rolled arms displayed at the two ends of the structure.

Some examples of current and future implementations of this technology are the three pairs of iROSA solar arrays that NASA is launching in the trunk of SpaceX Dragon 2 from early 2021 to early 2023 or the Power and Propulsion Element of the Lunar Gateway and Double Asteroid Redirection Test (DART) mission, which will use ROSA technology to power its solar electric propulsion [16][17].

Regarding these flexible solar panel technologies, it can be concluded that the application of these systems to Origami-based solar panel structures is not only a possible solution, but it can be the future for the CubeSat manufacturing industry and nanosatellite launching missions.

2.3 Attitude Determination and Control Subsystem

Every deployed spacecraft must be able to acquire, control, and maintain a particular orientation to fulfill the objective for which it has been designed. Firstly, in order to fulfill this need, it is of paramount importance to have an inertial frame or another reference, such as nearby objects or the light of some celestial body.
Therefore, it is a fundamental requirement to have a good enough measurement system to be able to recognize some of the already mentioned elements and, after the proper processing of this measured information, work over the required actuators to maintain and, if necessary, correct the orientation of the spacecraft to match the imposed attitude setting.

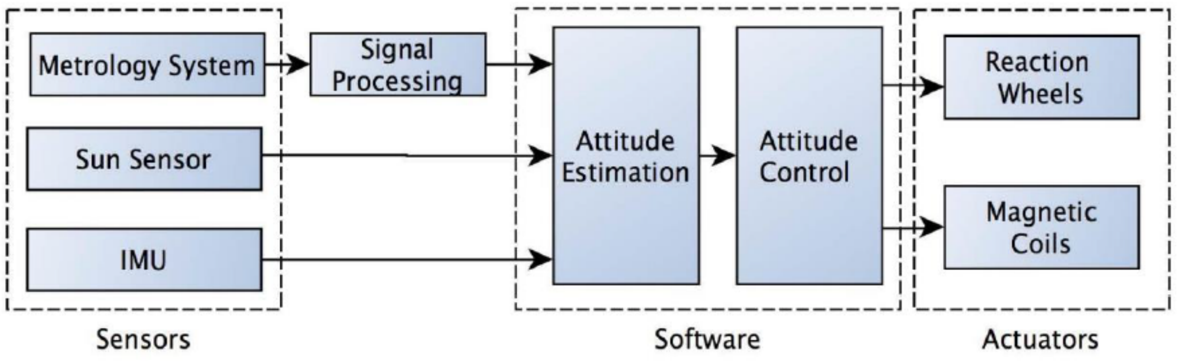


Figure 2.8 Block diagram of an attitude determination and control system. Source: [18]

Having said that, to provide a good enough introduction to the subsystems that will be studied in this project, the following devices will be briefly explained:

2.3.1 Microcontroller

According to figure 2.8, the software in charge of executing the corresponding control algorithms must be implemented in some hardware device that will be responsible of acting as an interface between the received inputs from sensors and the outputs in charge of powering the actuators.
The computational specifications of these devices can be diverse depending on the requirements of the implemented project. When talking about this particular project, the computational requirements of the control device are not a delimiting factor, instead, the versatility, simplicity, and compactness of the device are much more relevant, as well as the availability on the market and the cost of the hardware unit.

2.3.2 Inertial Measurement Unit

So as to know the angular speed, orientation, and angular acceleration of the satellite over time are of paramount importance to use an Intertial Measurement Unit (IMU). These devices mainly consist of three accelerometers and three gyroscopes (6 degrees of freedom), with which it is possible to measure the acceleration and angular speed of the system at all three axes. Additionally, to express the relative orientation, the IMU devices use navigation angles or Euler angles (Figure 2.9), which are represented as three orthogonal rotations around the X (roll), Y (pitch), and Z (yaw) axis.

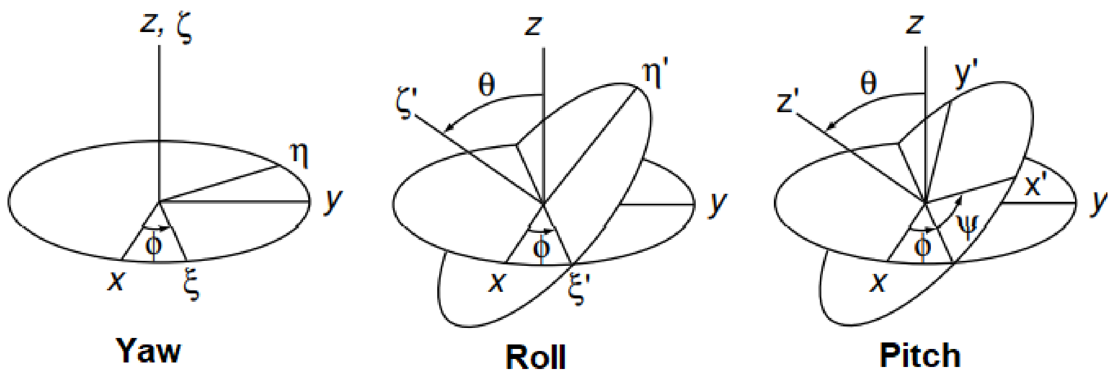


Figure 2.9 Sailing angles or Euler angles. Source: [19]

It is common to find IMUs with 9 degrees of freedom. The particularity in these cases is the additional incorporation of a 3-axis magnetometer or magnetic compass. Thanks to the magnetometer, it is possible to know at all times where the magnetic north of the Earth, or another celestial body, is situated with respect to the system. This can reduce cumulative errors that can be generated by using just the accelerometer and gyroscope calibrated to a specific reference.

2.3.3 Communication devices

During future tests with the Cubesat prototype that is going to be designed, it will be necessary to establish communications that allow the user to operate the system. In this way, the need to use a type of wireless communication that allows the user to establish the desired position instructions and read the data collected by the sensors comes into play. And a comparison of the main wireless communication technologies used for the transmission of information from sensors can be found at Table 2.1.

	ZigBee	**Sub-GHz**	**Wi-Fi**	**Bluetooth**
Physical layer standard	802.15.4	Owner/ 802.15.4g	802.11	802.15.1
Main application	- Monitoring - Control	- Monitoring - Control	Web, e-mail, video	Wired replacement
Battery life (days)	100-1000+	1000+	0.5-5	1-7
Network size	100s to 1000s	10s to 100s	32	7
Bandwidth (kbits/s)	20-250	0.5-1000	11000+	720
Range (m)	1-100+	1-7000+	1-30+	1-10+
Network architecture (most common)	Mesh	- Point by point - Star	Star	Star
Optimized for	- Scalability - Reliability - Low power demand - Low cost	- Long range - Low power demand - Low cost	Speed	- Low cost - Comfortable

Table 2.1 Main wireless communications for sensor information transmission. Source: Own

Nonetheless, the previous table 2.1 summarizes only the wireless communication technologies that could be applied to this project for the testing of the CubeSat in a laboratory setting. Consequently, these technologies would be unable to communicate with a CubeSat once in space, and it will be needed to substitute the communication module to translate the proposed solution to a final deployment setting. Considering so, and because of design specifications fixed at previous iterations that will be explained later on, the selected and used communication method is Bluetooth technology.

2.3.4 Magnetorquers

Along with reaction wheels, magnetorquers are the most common actuators to maintain or correct the orientation of satellites. This systems consist on a solenoid which, broadly speaking, allow an interaction with the surrounding magnetic field thanks to the magnetic dipole created by circulating current through its set of coils. As these elements are fixed on the satellite itself, the magnetic force exerted on the surrounding magnetic field generates an inverse magnetic force, resulting in a mechanical torque on the satellite.

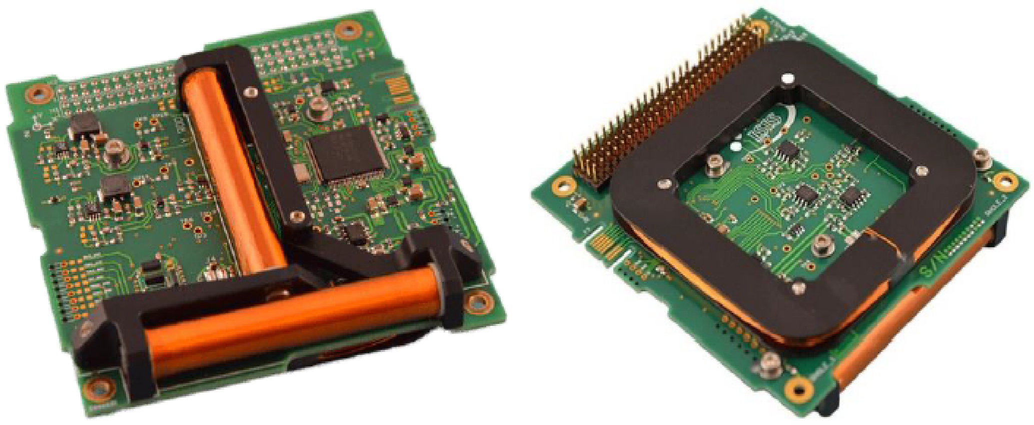


Figure 2.10 Magnetorquers card and three-axis magnetometer. Source: [20]

The main advantage of magnetorquers (Figure 2.10) is that, depending on the winding and the core used, it is possible to achieve an auxiliary attitude control actuator system with low consumption and reduced volume. Both features are well suited to the needs encountered in satellite design projects.
However, the present project does not contemplate the incorporation of any magnetorquer, since it is considered that, to achieve the objective of this study, the use of reaction wheels is a good enough solution.

2.3.5 Reaction wheel (RW)

Reaction wheels (Figure 2.11) are disks through which, by making variable angular speed changes, it is possible to control the attitude of a satellite thanks to the exchange of angular or kinetic momentum. They are usually used together with an electric motor that rotates the disk at the desired speed. These flywheels allow satellites to be oriented without the use of fuel-consuming propulsion systems.

When a jet wheel rotates, the angular momentum of the satellite that incorporates it is being modified. As the angular momentum of the assembly is conserved, this satellite suffers a change in its angular momentum in the opposite direction to the rotation of the jet wheel. With the use of three orthogonal reaction wheels, it is possible to orient the satellite with respect to any three-dimensional axis of rotation.

The main difference between flywheels and reaction wheels is that the former is designed to permanently rotate at a certain constant speed to provide gyroscopic stability, while reaction wheels frequently change their speed and direction to generate torques that control the attitude of the satellite.

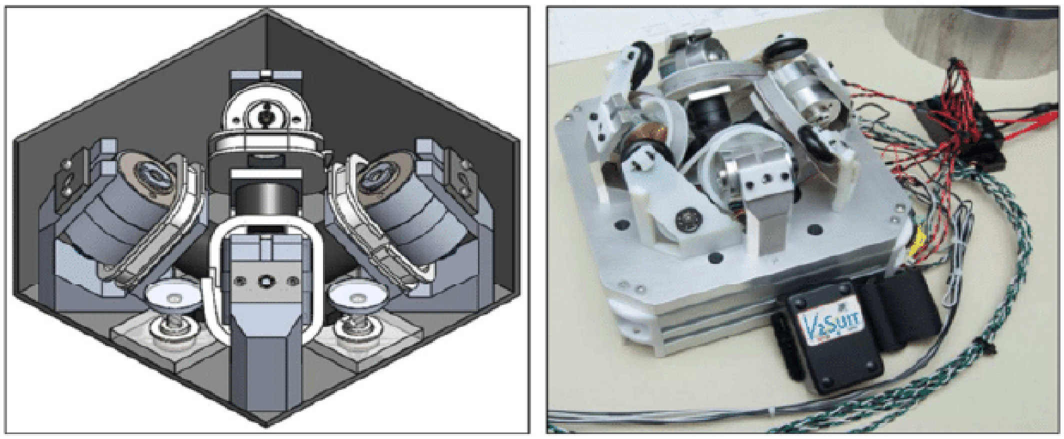


Figure 2.11 Example of tetrahedral configuration of RW (V2Suit). Source: [21]

2.3.6 Power supply systems

In real conditions, the power system of a satellite is based on the use of a battery that is recharged with the energy captured by photovoltaic panels (Figure 2.12). However, since the CubeSat designed for this project is intended to be used inside a test laboratory, this recharging source will be ruled out so as to focus all the efforts in the design of the Origami-based solar panel deployment mechanism.

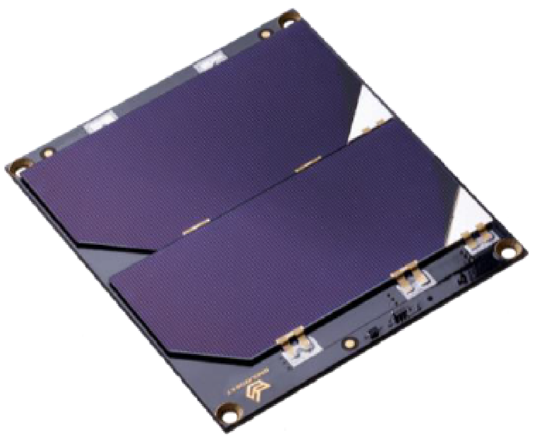


Figure 2.12 Solar panel used in 1U CubeSats real space applications. Source: [22]

Moreover, taking into account the high degree of complexity that the developed origami-based solar panel subsystem presents, both the displacement and the connection of these solar panels on the developed prototype will not be possible. Consequently, only the folding and unfolding mechanism and structure of this subsystem will be developed, and the solar panels will be substituted by a placeholder folded according to the starshade Origami pattern, that can be later replaced and connected to the satellite main system once the panels are completed by future projects of PLATHON team [3].

Chapter 3

Mechanics

3.1 Deployable solar panel subsystem

Taking into account the already seen CubeSat subsystems based on deployment mechanisms (see Deployable Structures 2.2), the following step will be to develop the first concept to find the deployment mechanisms that best fits the proposed prototype requirements.

In order to do so, the Starshade NASA project will be taken as the starting point in terms of origami pattern origin and will be thoroughly studied to give birth to the final deployment mechanism. To begin with this study, and taking into consideration the following image (Figure 3.1) extracted from the Starshade project, it has been deduced that the best way to develop an unfolding method for the proposed origami pattern shapes (see Figure 3.8) is to develop a mechanism that, by a radial extension movement of the vertexes of the external polygonal shape, would trigger the unfolding of the whole structure.

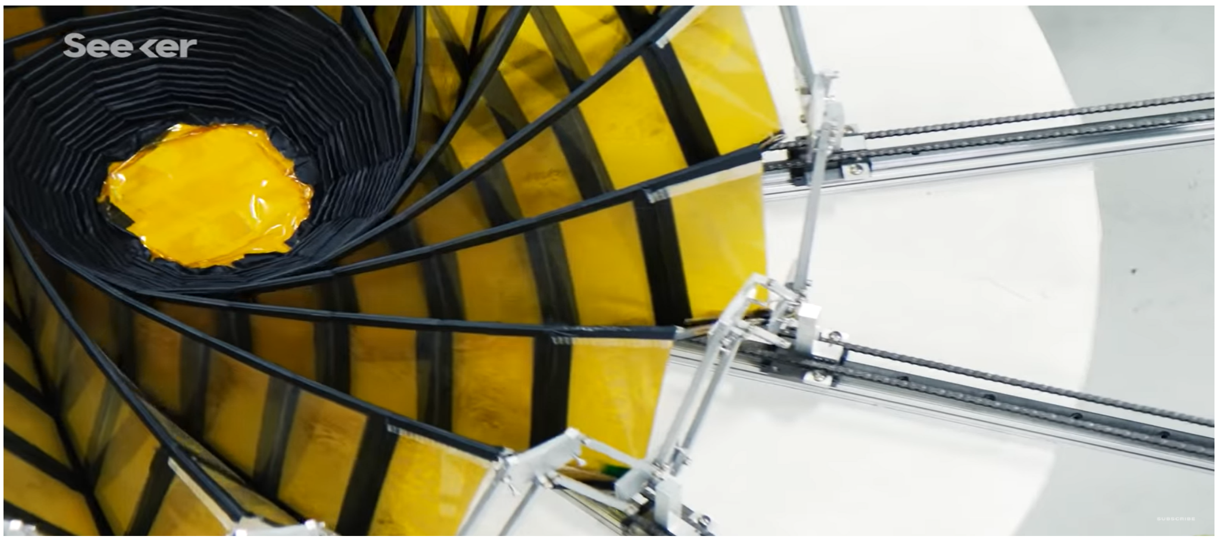


Figure 3.1 Starshade origami pattern demonstration. Source: [8]

Having said that, and after preliminary bibliography research, it has been decided that the deployment method used to develop the final deployable solar panel subsystem will be based on the concept of rhomboidal shaped deployable arms. The already mentioned deployable arm structure can be seen in figure 3.2, extracted from Omar Fabrisio Avellaneda Lopez's Ph.D. thesis [23].

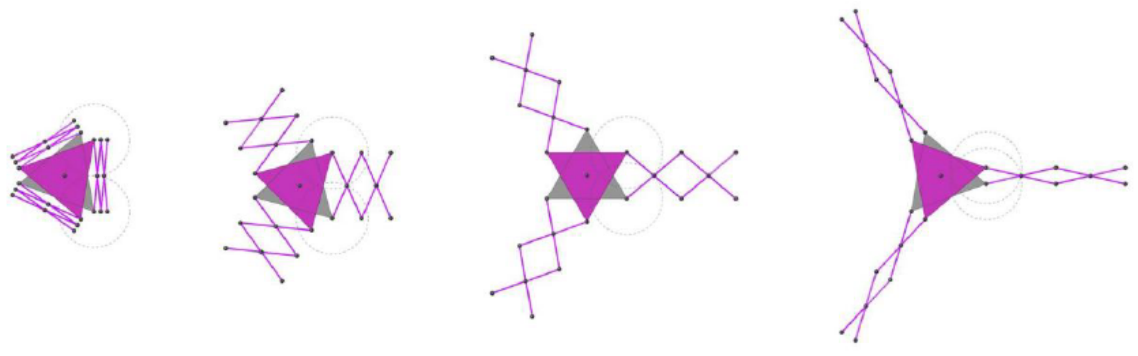


Figure 3.2 Deployable mechanism operating principle diagram. Source: [23]

Once a clear first deployable subsystem concept has been reached, the next step will be to begin the conceptual design process to provide the best possible solution for the proposed mechanism.

3.2 Deployable solar panel prototyping

The main objective of the following section of this chapter will be to study, justify, and explain the development and prototyping of the most mechanically related component evaluated during this project, which is the deployable origami-based solar panel subsystem. To ensure a better comprehension of the developed mechanisms, the section is divided into the following parts:

- **Design of preliminary prototype:** Firstly, a proper justification of the technical resources required to shape the final solution must be performed, dealing with matters such as used materials, optimal manufacturing methods, and an explanation of the mechanisms, which should be done in terms of proportion and design.
- **Design of the final prototype:** In the second place, after some testing of the first iteration of the proposed solution, some malfunction were found and, in this fragment of the chapter, these malfunctions will be identified, evaluated, and solved so as to present the final proposed solution to the subsystem studied in this chapter.

3.2.1 Design of preliminary prototype

Once selected a suitable candidate for the deployment mechanism, see Appendix A, a previous study of its technical viability will be required to ensure that the optimum solution for the already mentioned problems can be obtained. This study of viability will be divided into the following parts:

3.2.1.1 Manufacturing material selection

To begin with, the determination of the material which offers the most convenient properties to fulfill the mechanism requirements will be of paramount importance in the designing process. These materials can be divided depending on whether they are aimed towards prototyping purposes or final design for launching proposes.

On the one hand, when talking about materials for commercial and final launching purposes, it must be mentioned that the most used materials for CubeSat frames manufacturing applications are the ones presented in the following table (Figure 3.3):

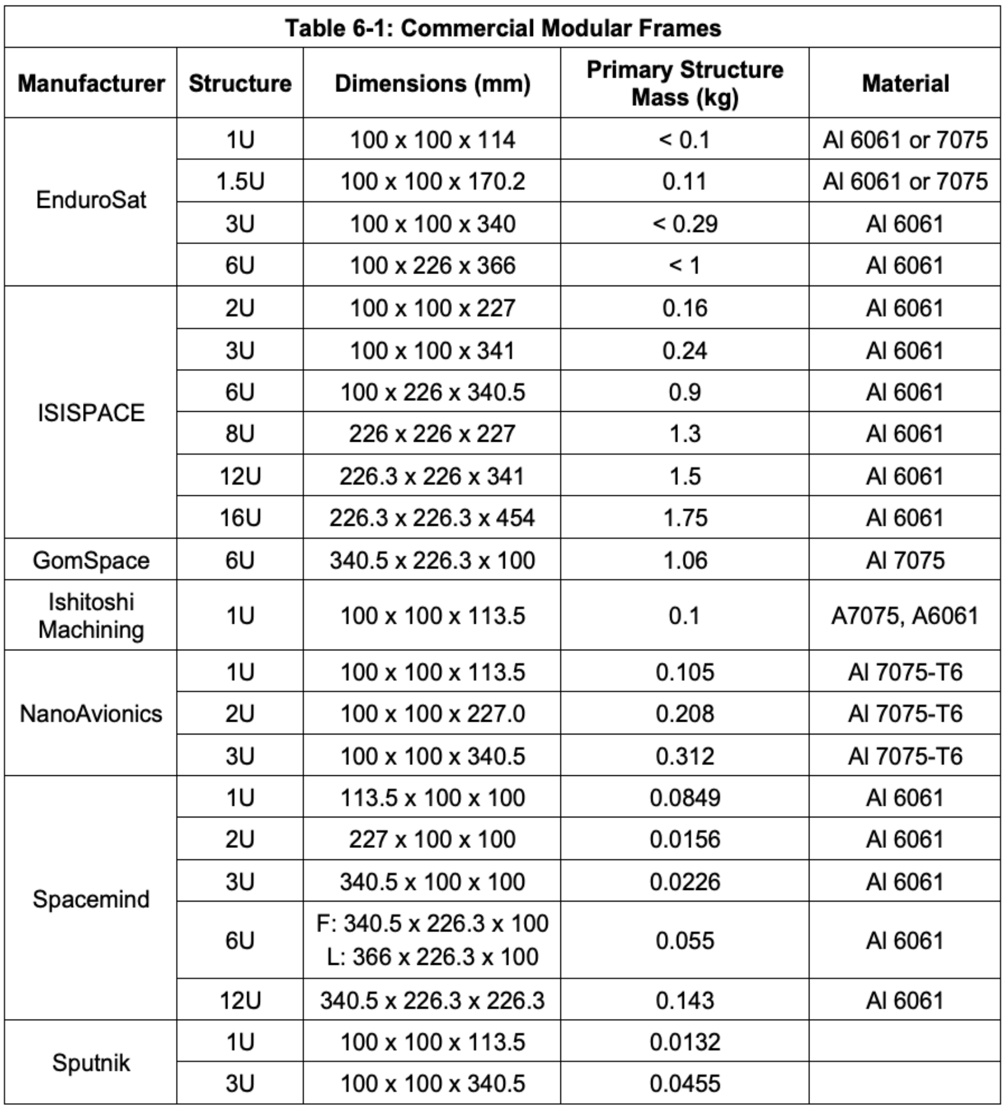


Table 6-1: Commercial Modular Frames

Manufacturer	Structure	Dimensions (mm)	Primary Structure Mass (kg)	Material
EnduroSat	1U	100 x 100 x 114	< 0.1	Al 6061 or 7075
	1.5U	100 x 100 x 170.2	0.11	Al 6061 or 7075
	3U	100 x 100 x 340	< 0.29	Al 6061
	6U	100 x 226 x 366	< 1	Al 6061
ISISPACE	2U	100 x 100 x 227	0.16	Al 6061
	3U	100 x 100 x 341	0.24	Al 6061
	6U	100 x 226 x 340.5	0.9	Al 6061
	8U	226 x 226 x 227	1.3	Al 6061
	12U	226.3 x 226 x 341	1.5	Al 6061
	16U	226.3 x 226.3 x 454	1.75	Al 6061
GomSpace	6U	340.5 x 226.3 x 100	1.06	Al 7075
Ishitoshi Machining	1U	100 x 100 x 113.5	0.1	A7075, A6061
NanoAvionics	1U	100 x 100 x 113.5	0.105	Al 7075-T6
	2U	100 x 100 x 227.0	0.208	Al 7075-T6
	3U	100 x 100 x 340.5	0.312	Al 7075-T6
Spacemind	1U	113.5 x 100 x 100	0.0849	Al 6061
	2U	227 x 100 x 100	0.0156	Al 6061
	3U	340.5 x 100 x 100	0.0226	Al 6061
	6U	F: 340.5 x 226.3 x 100 L: 366 x 226.3 x 100	0.055	Al 6061
	12U	340.5 x 226.3 x 226.3	0.143	Al 6061
Sputnik	1U	100 x 100 x 113.5	0.0132	
	3U	100 x 100 x 340.5	0.0455	

Figure 3.3 Materials used for CubeSat frame manufacturing. Source: [24]

As it can be easily seen in the previous table, the most used material for CubeSat mechanical applications is aluminium, being commonly used both Al 6061 or Al 7075-T6.

On the other hand, considering the material selection for prototyping purposes, which is the main objective of this project, it is necessary to talk about polymeric materials, which will be a more versatile election for prototype manufacturing purposes.

In the following table 3.1, it will be represented a comparative view of different plastic materials that can be used for prototyping proposes (also depending on their manufacturing processes):

Material	**Printing technology**	**Properties**	**Application**
ABS	FDM Blinder Jetting SLA Poly Jetting	- Mechanical resistance - Lightness - Reliability - Flexibility - Thermal resistance	- Architectonic modeling - Conceptual modeling - Diy applications - Parts manufacturing
PLA	FDM SLA SLS	- Biodegradability - Food safe	- Conceptual modeling - Diy applications - Functional modeling - Functional parts manufacturing
PA12	FDM SLA	- Resistance and flat surface - Chemical resistance - Slightly flexible - Thermal resistance	- Conceptual modeling - Functional modeling - Medical applications - Tool manufacturing - Visual arts
PET	FDM	- Mechanical resistance - Food safe - Flexible - Flat surface	- Diy applications - Functional modeling - Food industry applications

Table 3.1 PET, PLA, ABS, PA12 Comparative table. Source: Own

Having said that, and considering the material properties already mentioned, it is time to perform a deeper study of the most suitable manufacturing methods so as to select the most convenient one, also considering the already introduced materials.

3.2.1.2 Manufacturing method selection

Firstly, keeping in mind the complex geometries that will be displayed on the designed pieces and the limited amount of time that can be spent in prototyping, it must be said that the manufacturing method that will best fit the time and budget restrictions of this project is based on additive manufacturing (3D printing).

Taking this into account, and considering the available additive manufacturing technologies in UPC facilities, two different 3D printing methodologies must be mentioned:

- **FDM 3D printing technology:**
  FDM (Fused Deposition Modeling) printing technology is based on the deposition of a plastic fused wire to create a mesh configuration that (layer by layer) forms the desired final object.
  The simplicity of this method makes this printing technology one of the most used ones worldwide.
- **HP MJF 3D printing technology:**
  HP MJF (Multi Jet Fusion) additive manufacturing technology is based on the deposition of plastic dust layers and their subsequent curing to shape the final solid part. Considering so, the curing process is performed by a combination of a curing resin that link the dust particles, and an external UV source that solidify the mixture .
  Some of the main advantages of the use of this printing method are the compactness of the final printed part (considering that the final parts are nearly sintered) and the surface finish of the final part.

After the previous description of each additive manufacturing technology, the following pros and cons table 3.2 is presented to make the final decision about which of these methods will be used at the prototype.

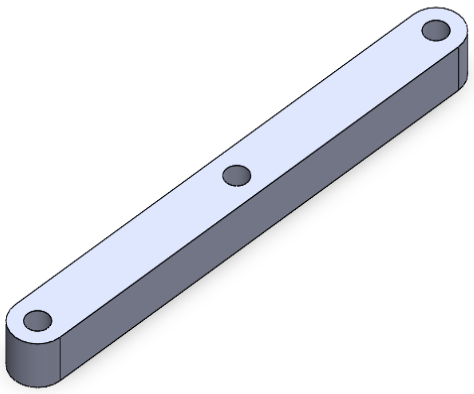
	Pros	**Cons**
FDM	- Ideal for prototyping - Ideal for small production - Low cost - Produces strong functional models	- Non-isotropic parts result in weaker parts - Fused deposition leads to more wastes - Non easily recyclable materials
HP MJF	- True volume manufacturing (low wastes) - Higher quality finished - More cost-effective for larger print volumes - Isotropic pieces that provide uniform mechanical properties	- Lower material availability - Lower color availability - Higher production costs if used for small productions

Table 3.2 Additive Manufacturing Pros and Cons table. Source: Own

Regarding the already mentioned characteristics of each technology, the selected 3D printing method will be FDM additive manufacturing. Moreover, it is observed that FDM printing technology is much more accessible than any other 3D printing mechanism, and it can also better achieve the time and quality requirements due to the easiness of the iteration process. Consequently, the used 3D printer for the manufacturing of all the mechanical parts of this project will be the **Original Prusa i3 MK3S+**.

3.2.1.3 Deployable arm length determination

After the proper determination of the optimal material and manufacturing technology for the prototyping of the previously studied deployable solar panel subsystem, the prototyping process will begin with the determination of the dimensional characteristics of the most simple but relevant piece in the whole assembly, the deployable arm structure rod (Figure 3.4a).


(a) Deployable arm rod.

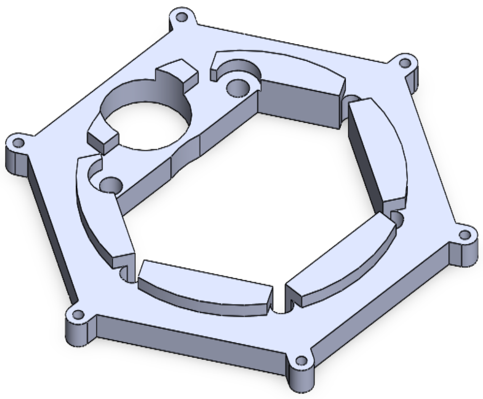

(b) Deployable subsystem base hexagon.

Figure 3.4 Deployable subsystem fundamental parts. Source: Own

Taking into consideration that the assembly must guarantee proper compactness when folded (despite pieces width being forcefully increased to provide enough rigidity to the plastic structure), the determination of the rod length will be closely related to the shape and size of the central hexagon of the whole set, which has been shown on the previous image (Figure 3.4b).
To successfully determine the proper dimensions of these pieces, the limitations imposed for 1U CubeSats dimensional restrictions (10x10x10cm) has been considered.

Taking then advantage of the geometrical calculations provided by the SolidWorks software, it has been concluded that the best configuration for the central piece is a 45mm side length hexagon (which allows a long enough bar length while keeping proper compactness of the system).
However, the possible collisions between the different arm pieces when folding the whole mechanism must be considered, and the final length of the independent arm rods is made slightly shorter than the side length of the hexagon, to ensure no collision is possible when the deployment system is fully closed. Therefore, a rod longitude of 40mm is selected and the final rod geometry becomes the following (Figure 3.5):

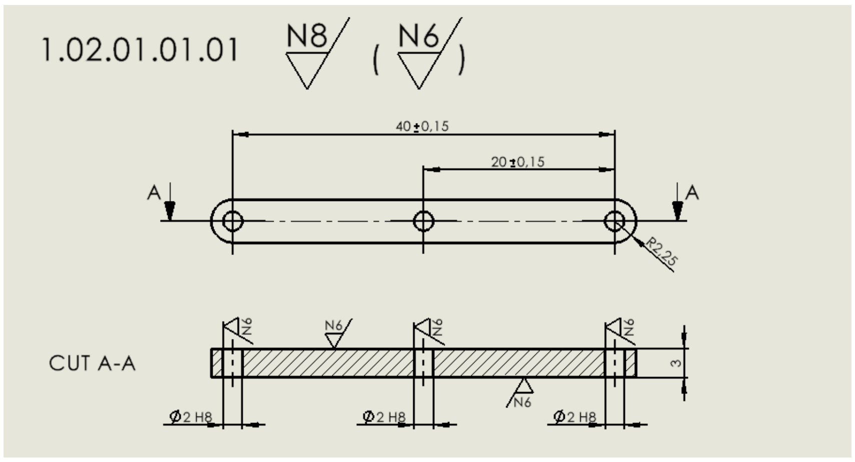


Figure 3.5 Deployable arm rod blueprint extract. Source: Own

Once a proper determination of the rod's longitude has been done, it is now time to determine the thickness of these rods to guarantee their integrity when being subjected to a torque. To do that a static test will be performed by using SolidWorks simulation software.
In order to do so, the first step is to determine the torque to which the studied rods will be subjected, and a previous study of the selected torque generator (stepper motor) will be required (Table 3.3):

Stepper motor 28BYJ-48 Parameters	
Model	28BYJ-48
Rated voltage (V)	5VDC
Number of phase	4
Speed variation ratio	1/64
Stride angle	5,625º/64
Frequency (Hz)	100Hz
DC resistance	$50\Omega \pm 7\%(25°C)$
Idle In-traction Frequency (Hz)	>600Hz
Idle Out-traction Frequency (Hz)	>1000Hz
In-traction Torque (mN· m)	>34,3mN· m (120Hz)
Self-positioning Torque (mN· m)	>34,3mN· m
Friction torque (gf· cm)	600-1200 gf· cm
Pull in torque (gf· cm)	300 gf· cm
Insulated resistance	$>10M\Omega(500V)$
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in temperature	<40K(120Hz)
Noise (dB)	<35dB(120Hz, No load, 10cm)

Table 3.3 28BYJ-48 Stepper motor's Parameters. Source: Own

After seeing the required motor data (Table 3.3), and considering the fundamental gear dimensions shown in the table 3.5, the forces to which the rods are submitted are studied and calculated.

$$\frac{F_{motor}}{(d_{gear} + l_{gear-rod})} = F_{rod} \tag{3.1}$$

$$F_{motor} \approx 34,3mN \cdot m = 34,3N \cdot mm \tag{3.2}$$

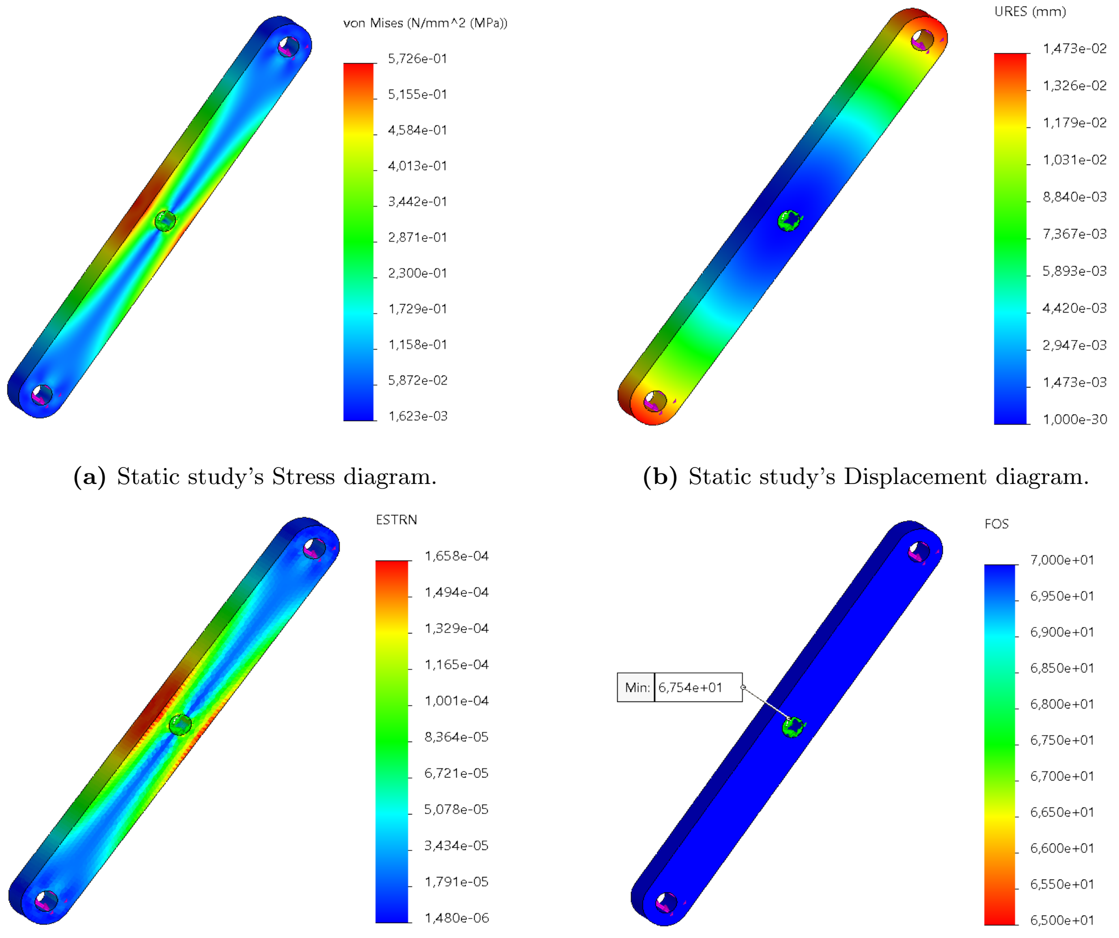
$$d_{gear} = \frac{15,5mm}{2} - \frac{2,2mm}{2} = 6,65mm \tag{3.3}$$

$$l_{gear-rod} = 45,04mm - \frac{69,4mm}{2} + \frac{2,2mm}{2} = 11,44mm \tag{3.4}$$

$$\frac{34,3N \cdot mm}{(6,65mm + 11,44mm)} = F_{total} = 1,896N \tag{3.5}$$

$$F_{rod} = \frac{1,896N}{6} = \textbf{0,316 N} \tag{3.6}$$

Therefore, the static study of the arm rods is performed, obtaining the following results (Figure 3.6):



(a) Static study's Stress diagram.

(b) Static study's Displacement diagram.

(c) Static study's Strain diagram.

(d) Static study's Factor Of Safety diagram.

Figure 3.6 Deployable arm rod's static study diagrams. Source: Own

As a consequence, the following table can be represented so as to summarize the obtained results and to ensure a proper understanding and evaluation of this study (Table 3.4):

Piece	1.02.01.01.01	
Exerted force (N)	F= 0,316	
Force diagram	Figure 3.7	
Failure criteria	VonMises	
Mechanical test	**Min**	**Max**
Stress (N/mm^2)	$1,623 \cdot 10^{-3}$	$5,726 \cdot 10^{-1}$
Displacement (mm)	$1 \cdot 10^{-30}$	$1,473 \cdot 10^{-2}$
Strain	$1,480 \cdot 10^{-6}$	$1,658 \cdot 10^{-4}$
Factor of Safety (FOS)	68	-

Table 3.4 Deployable arm rod's static study results. Source: Own

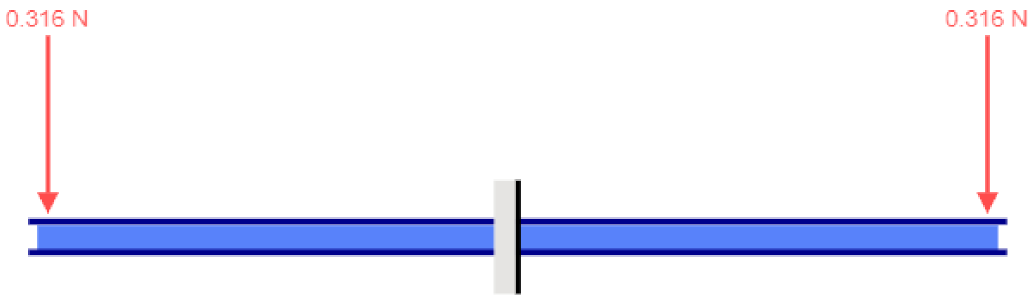
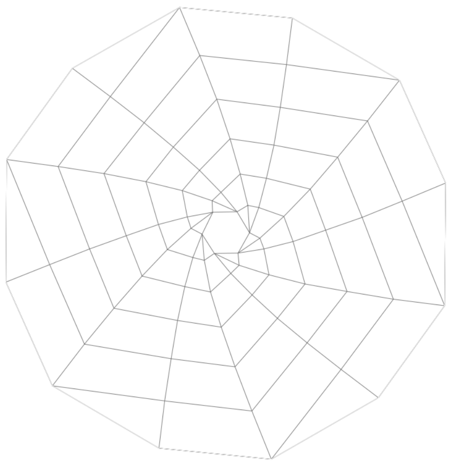


Figure 3.7 Rod's static test beam diagram. Source: Own

After seeing the previous results (Table 3.4), more specifically the safety factor value, it can be concluded that the design of this particular part can fulfill the imposed torque requirements presenting a rod thickness of 3mm.

3.2.1.4 Solar panel sizing

The next step in the design process will be the determination of the most suitable shape and dimensions of the origami-based solar panel that will be attached to the deployable mechanism. In order to study and guarantee the highest possible level of compactness, the two following origami patterns have been studied (Figure 3.8):


(a) Starshade origami pattern (12 plates).

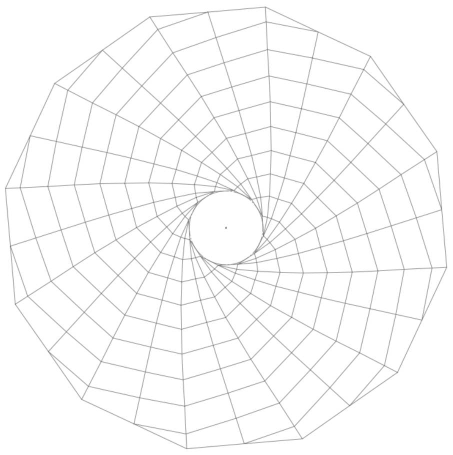

(b) Starshade origami pattern (24 plates).

Figure 3.8 Studied Starshade origami patterns. Source: Own

It must be mentioned that the main difference between the already seen origami patterns lies in the fact that, once they have been folded, the level of compactness that they present is different, being the one presented at Figure 3.8b more compact when folded. This is because the higher the number of "triangular" surfaces of the pattern, the higher the compactness when it is folded.

In spite of the previous fact, other factors such as the force required for unfolding must be considered, and the pattern presented at Figure 3.8a is ultimately found as the arrangement that best fits the project requirements.

Once the previous conclusion is reached, and with the main objective to properly dimension the origami-based solar panel structure, the following considerations and limitations must be kept in mind.

- The design of this deployable solar panel system is focused on the improvement of already studied solar panel mechanisms used at previous CubeSat designs, such as the following wall-integrated solar panel (Figure 3.9) that proposes the inclusion of 6 independent panels covering the CubeSat external surface:

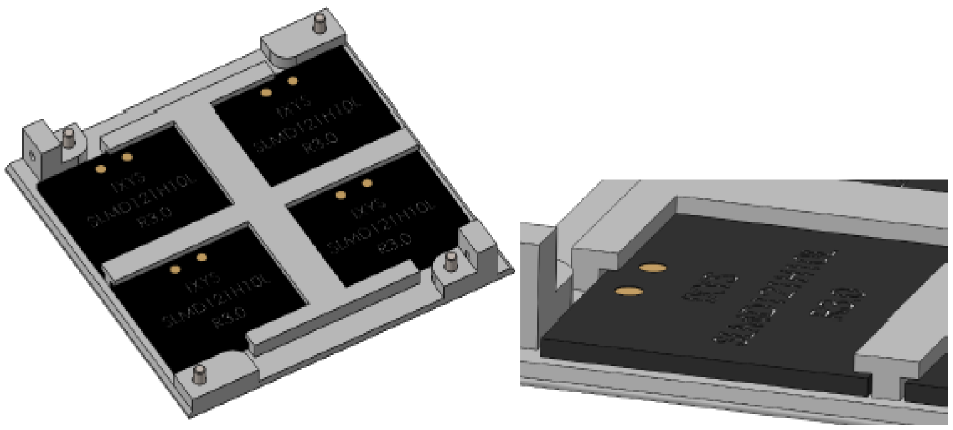
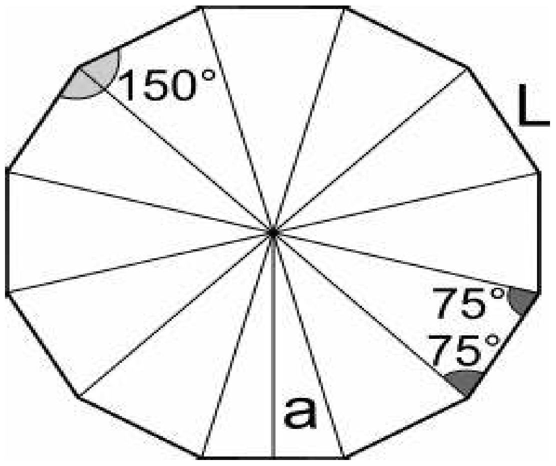


Figure 3.9 Wall-integrated solar panel subsystem. Source: [25]

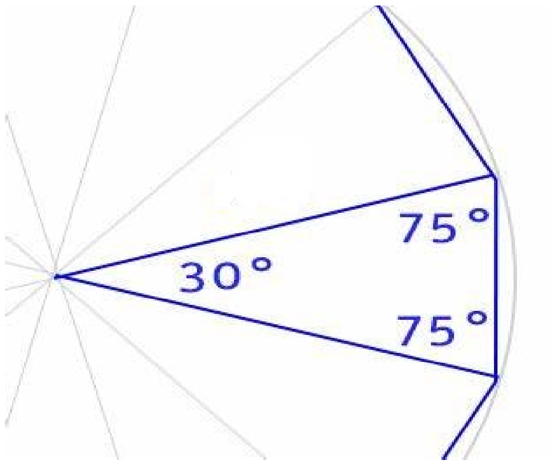
- The dimensioned solar panel must fit an arm's length of 220mm because of the limitations and restrictions of the selected compact arm design.
- Given it is pursued to achieve an enhancement of the effective solar reception crossection of the solar panels, the size of the fully extended deployable panel is set equal to the surface of a completely unfolded cube (i.e. six times the cube's wall surface). That way, a single origami panel will be able to generate equivalent power to the fully covered CubeSat obtained when the system at Figure 3.9 is used.

$$A_{pattern} \equiv A_{cube} = 100 \cdot 100 \cdot 6 = 60000mm^2 \tag{3.7}$$

Beginning now the final size calculation process, and in order to better understand the factors that will be used in the following formulas, a representation of the geometrical considerations taken into account has been exemplified as follows (Figure 3.10):



(a) Dodecagon's geometrical parameters diagram.



(b) Dodecagon's angles diagram.

Figure 3.10 Dodecagon's geometrical parameters. Source: [26]

$$\begin{cases} 12 \cdot a \cdot \frac{L}{2} = 60000mm^2 \rightarrow L = \frac{10000mm^2}{a} \\ \tan 75 = \frac{a}{(L/2)} \rightarrow L = \frac{2a}{\tan 75} \end{cases} \rightarrow \frac{2a}{\tan 75} = \frac{10000mm^2}{a} \rightarrow \mathbf{a} = \mathbf{136{,}60mm} \tag{3.8}$$

However, once this calculation is completed, it is found that the proposed arm extension mechanism can tolerate a much bigger solar panel without suffering any deployment issue, and it is decided to duplicate the surface area of the panel to obtain a final **a** = **193,19mm** which will allow $\mathbf{A_{pattern}} = \mathbf{120000mm^2}$.

Once reached the previous configuration for the solar panel, and returning to the main goal of this section, which is to increase the power output that was reported to be produced by the wall-integrated solar panel proposed by Antonio Bonilla Lopez [25] by increasing the incision surface, an estimation of the improvement in terms of power generation must be also considered.

Considering so, it must be accounted that the solar flux in space and outside Earth's shadow cone is approximately 10 times higher than the long-term average at the surface of spinning [27]. It has been then assumed that in the Low Earth Orbit, the solar flux is approximately 1350 W/m2 [28], but the real power output of any solar panel in this conditions will be deeply affected by the reliability of this assumption.
On the other hand, the photovoltaic panels that are commonly used for CubeSats are composed of III-V multi-junction solar cells, because of their high-power conversion efficiencies [29], as well as their ability to outperform other solar power generation systems when exposed to extraterrestrial solar spectrum. As a consequence of this selection, an outstanding efficiency close to 30% quantum yield is observed, and a reduction of the solar panel area requirement is produced in comparison to other solar panel systems.

These photovoltaic panel specifications also exemplify the great improvement in terms of power generation that these devices can experience if the evolution in these technologies leads to an efficiency enhancement.

Therefore, and considering that the original system can only effectively expose one of its facets to light while the origami pattern is designed to be directed towards the sun to expose its full surface, the final power that the discussed systems can supply, assuming the average solar flux and efficiency previously exposed, is calculated as follows:

$$\begin{cases} A_{original} = 60000mm^2 \xrightarrow{16\% exposure} A^{eff}_{original} = 0,01m^2 \\ A_{pattern} = 120000mm^2 \xrightarrow{100\% exposure} A^{eff}_{pattern} = 0,12m^2 \\ \eta = 0,3 \\ \dot{q}" = 1350W/m^2 \end{cases} \xrightarrow{P=A\cdot\dot{q}"\cdot\eta} \begin{cases} P_{original} = \mathbf{4{,}05W} \\ P_{original} = \mathbf{48{,}60W} \end{cases} \tag{3.9}$$

As a consequence, the comparison between the the two previously determined results show the following optimal percentage increment in energy production, derived from shift from wall-integrated panels to origami deployable system:

$$\frac{48,60W}{4,05W} \cdot 100 = \mathbf{1200\%} \tag{3.10}$$

As it was expected, the energy production for the origami-based deployable solar panel configuration is twelve times higher than the original wall-integrated solar array.
Once the power generation study has been concluded, it is also important to point out that the selected solar panel topology allows great improvements in terms of surface area requirements, since, as later demonstrated, minor design changes on the dodecagonal geometry and its radial unfolding method can be translated to a meaningful increase on the solar panel area.

To demonstrate the previous statement, a greater study about the relationship between the apothem of a dodecagon and its surface is required. To do that, the following mathematical expression has been deduced (Equation 3.11):

$$A_{dodecagon}(mm^2) = 6 \cdot \frac{2a^2}{\tan 75} \tag{3.11}$$

Once the previous mathematical expression was obtained, the following graphic representation was carried out, which represents the relationship between the apothem of a dodecagon and its surface (Figure 3.11):

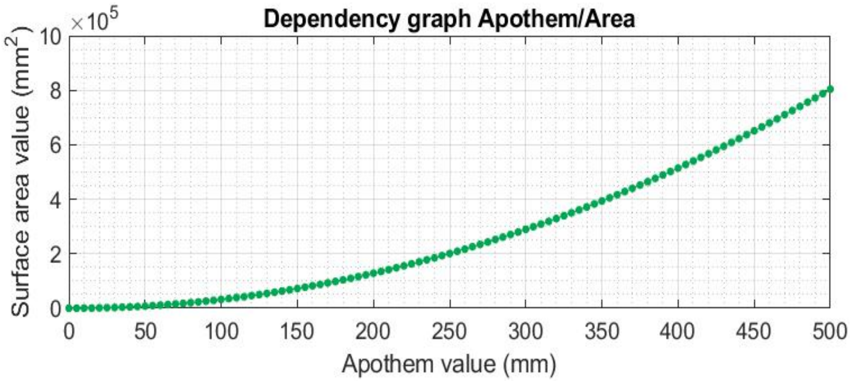


Figure 3.11 Dependency graph Apothem/Area. Source: Own

So as to also exemplify the dependency of the apothem value over the power generation, the graphic below has been also generated (Figure 3.12):

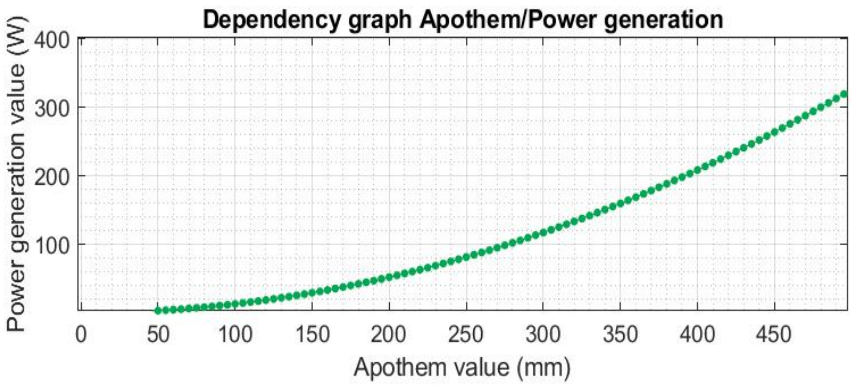


Figure 3.12 Dependency graph Apothem/Power generation. Source: Own

Regarding the result of this study, it can be concluded that the use of this deployable structure typology enables the designer to better adjust the surface of the solar panel to the power consumption requirements without great changes in the mechanism and CubeSat frame designs.

3.2.1.5 Torque transmission mechanism development

As the project moves on, it is necessary to do a more detailed explanation of the chosen torque transmission mechanism in charge of the deployment of the already established origami-based solar panel.

After the conceptual design process explained in the corresponding Appendix A, it has been concluded that the torque transmission mechanism will be based on a single-gear system in charge to transmit the movement of an implemented stepper motor, consequently generating a counterrotation movement of two hexagonal parts connected to the rhomboidal shape deployable arms (as it is shown in Figure 3.2). To clarify the explanation of this mechanism, the following graphical representation is added (Figure 3.13):

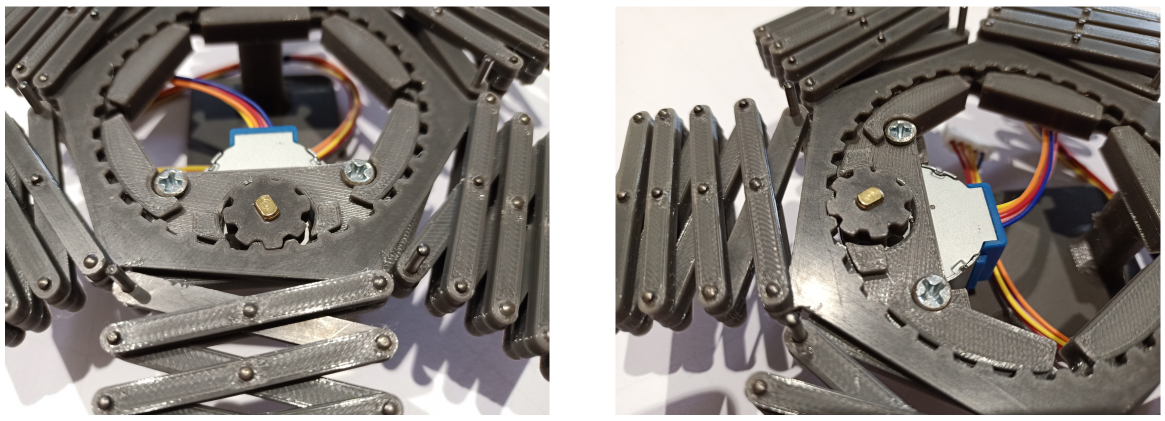

(a) Preliminary prototype gear mechanism front view. **(b)** Preliminary prototype gear mechanism side view.

Figure 3.13 Preliminary prototype gear mechanism. Source: Own

Once the introduction of the torque transmission mechanism has been included, a more detailed description of the gear-based device must be properly defined and justified. To do that, the following geometrical characteristics of the gears must be taken into account (Figure 3.14):

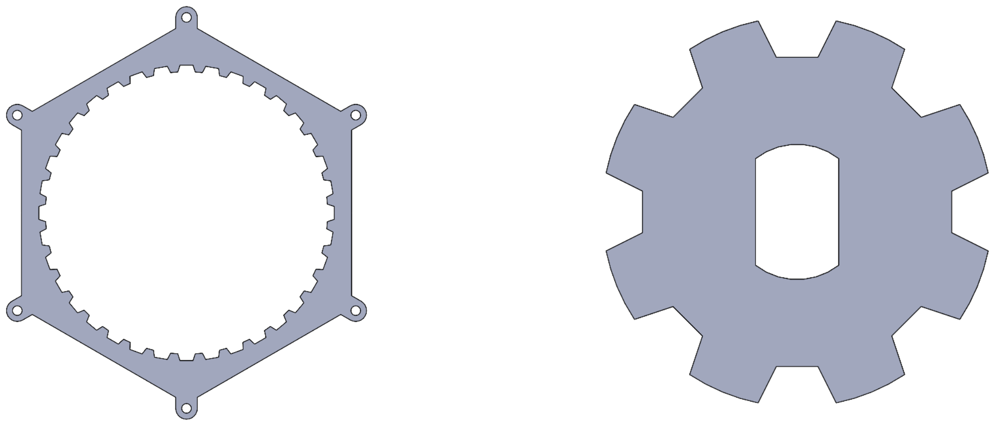

(a) Preliminary crown gear 3D model. **(b)** Preliminary pinion gear augmented view.

Figure 3.14 Preliminary gear mechanism fundamental parts. Source: Own

As it can be seen in the previous images (Figure 3.14), the chosen gear teeth geometry is trapezoidal, this geometry was chosen because of the simplicity that this offers when applied to prototypes that can be modified multiple times during the iteration process.

Considering some other characteristics shown in previous images, another relevant factor is the number of teeth that these pieces present (more specifically, the crown gear in figure 3.14a) which will be closely related to the transmission ratio of this mechanism.

The following table 3.5 represents the fundamental dimensions of both pinion and crown gear designed during the first iteration of the prototyping process:

	Crown gear	**Pinion gear**
Outside diameter, Do (mm)	65	15
Root diameter (mm)	68	12
Whole depth, ht (mm)	1,5	1,5
Circular pitch, p (mm)	5,67	5,89
Module or modulus, m	1,75	1,75
Number of teeth, N	36	8
Reduction ratio	4,5/1	

Table 3.5 Preliminary Gear fundamental dimensions. Source: Own

So as to simplify the calculations related to the rotation of the stepper motor needed for the proper deployment of the whole mechanism, the crown gear in figure 3.14a was designed to have 36 gear-teeth and present a rotation ratio of 10^{o}/tooth, while the pinion of figure 3.14b gear presents 8 gear-teeth, which implies a reduction ratio of the two gear system of 1/4,5.

3.2.1.6 Solar panel rotation mechanism

Once the already seen detailed explanation of every single mechanism in the solar panel deployable system prototype has been performed, the effect of the movement of this mechanism over the origami-based solar panel must also be considered.
Keeping in mind that solar array is based on an origami pattern (Figure 3.15), a rotation of the central hexagon during the unfolding of the origami structure can be expected.

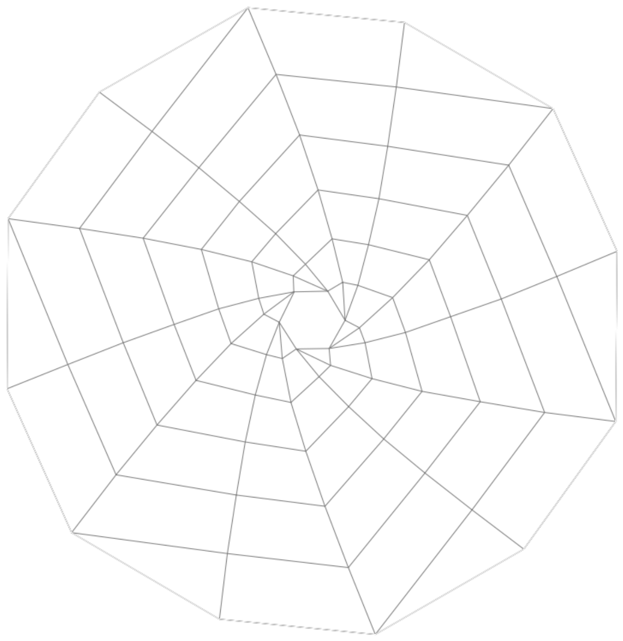
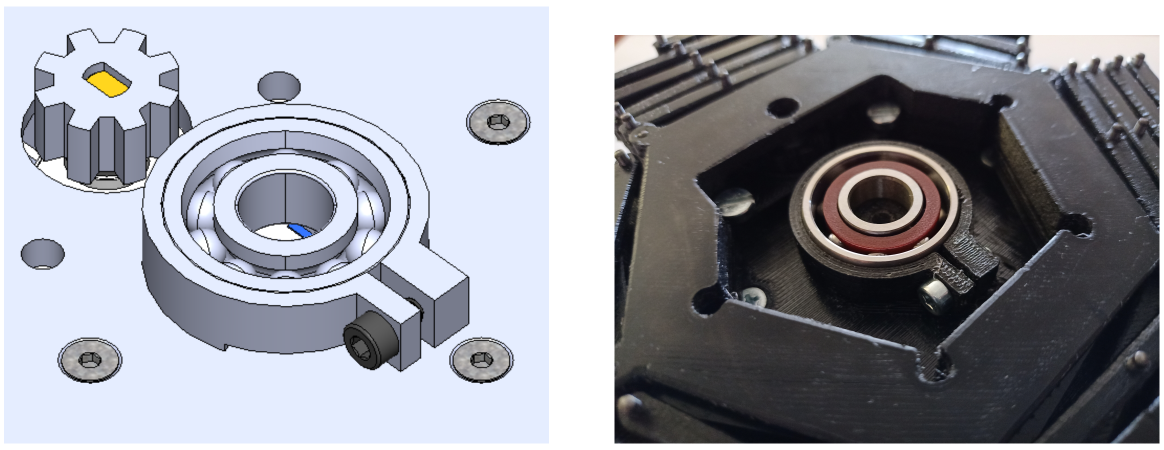


Figure 3.15 Selected Starshade origami pattern. Source: Own

Therefore, this rotation movement must also be considered and integrated into the prototype designs. In order to integrate this rotation movement into the prototype, a rotation mechanism has been implemented in the superior wall of the CubeSat, as can be seen in the figure 3.16:



(a) Wall integrated bearing 3D model. **(b)** Wall integrated bearing real prototype.

Figure 3.16 Wall integrated bearing system. Source: Own

The previous mechanism is therefore assembled by taking advantage of the elastic properties of the material that is being used to prototype (PET), which allows a bearing to be fixed to the CubeSat wall (Figure 3.16). By doing this, a platform can be attached to this bearing and, by subsequently fixing the origami-based solar panel to this platform, the rotation of the whole panel will be allowed (Figure 3.17).

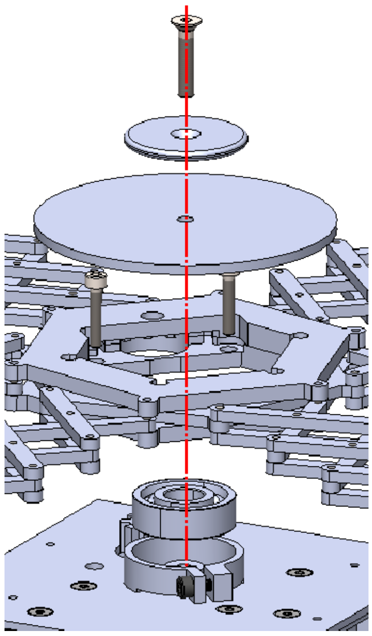


Figure 3.17 Rotating platform mechanism exploded view. Source: Own

So as to properly select the optimum bearing for this application, the following limitations must be considered:

- The stress to which the bearing will be submitted will be, at the same time, radial (to fix the bearing to the CubeSat) and axial (because of the weight of the platform and the panel).
- The bearing must be as small as possible in order to fulfill the project requirements.

After mentioning these requirements, it has been concluded that the selected bearing will be the following (Figure 3.18):

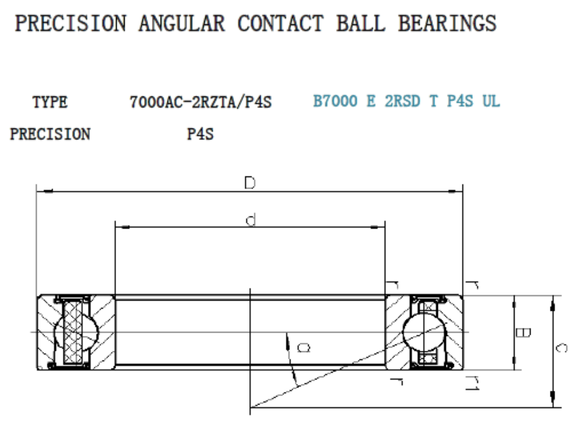


Figure 3.18 Precision 7000AC selected angular contact ball bearing. Source: Appendix F.7.1

3.2.2 Design of the final prototype

Once the preliminary prototype has been manufactured and tested, some malfunctions have been found in the torque transmission mechanism previously designed and justified.
These technical issues were produced because of the deformation of the toothed ring during the deployment of the mechanism, which lead to the hindrance of the movement of the toothed ring, making it impossible to deploy. To solve these problems, the following improvements were implemented:

3.2.2.1 Improvements over the first prototype

3.2.2.1.1 Redesign of the gear mechanism:

Taking into account the fact that the most closely related parts to the previously described malfunction of the first iteration of prototyping are the deployment gear-based mechanism parts, these will be submitted to a most detailed and exhaustive study.
So as to develop a suitable solution to the deformation issues, some modifications have been implemented in the previous crown and pinion gears, which have been summarized in the table 3.6:

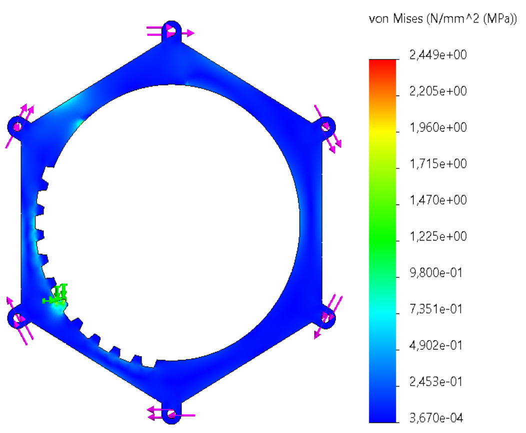
	Crown gear	**Pinion gear**
Outside diameter, Do (mm)	65	~~15~~ $\longrightarrow$15,5
Root diameter (mm)	~~68~~ $\longrightarrow$ 69,4	~~12~~ $\longrightarrow$ 11,1
Whole depth, ht (mm)	~~1,5~~ $\longrightarrow$ 2,2	~~1,5~~ $\longrightarrow$ 2,2
Circular pitch, p (mm)	5,67	~~5,89~~ $\longrightarrow$ 6,09
Module or modulus, m	~~1,75~~ $\longrightarrow$ 2	~~1,75~~ $\longrightarrow$ 2
Number of teeth, N	36	8
Reduction ratio	4,5/1	

Table 3.6 Final Gear fundamental dimensions modifications. Source: Own

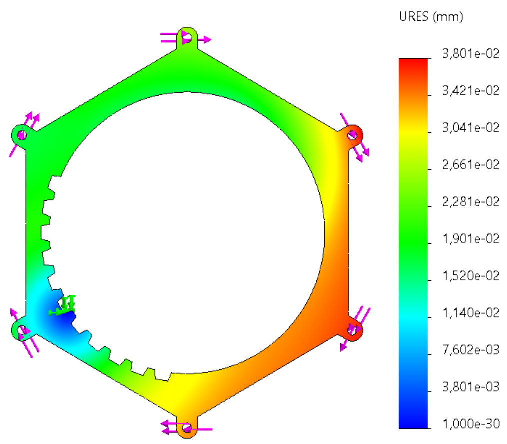
The demonstration of the technical viability of these implemented modifications will be probed by executing some computational tests over the modified 3D models designed using the mechanical CAD software SolidWorks.

The previously mentioned computational tests will be implemented aiming to achieve different goals depending on the tested pieces, and their results can be summarised as follows:

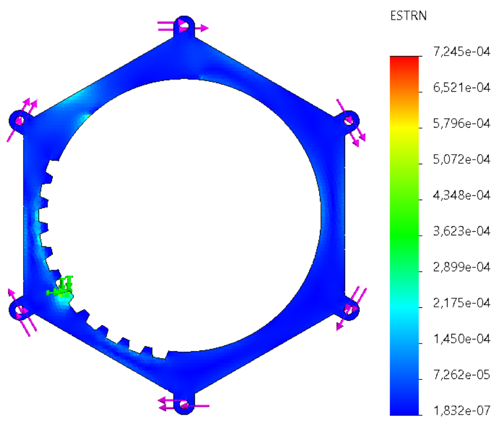
- **Tests of modified gear-based mechanism pieces:** The main goal of the implementation of this tests is to ensure the correct mechanical behavior of the gear teeth during the deployment maneuver, while maintaining a trapezoidal tooth geometry, which enables the designer to freely and easily modify its dimensions in order to fit the space availability requirements. Once the computational tests have been simulated, the obtained results are (Figure 3.19):



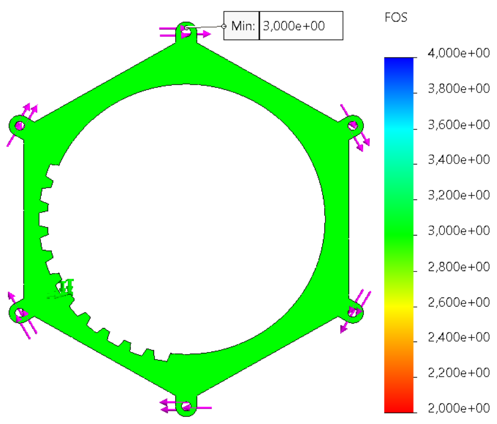
(a) Static study's Stress diagram.



(b) Static study's Displacement diagram.



(c) Static study's Strain diagram.



(d) Static study's Factor Of Safety diagram.

Figure 3.19 Modified Crown gear's static study diagrams. Source: Own

As it has been done in previous sections, to ensure proper comprehension of the results, the summary table presented below (Table 3.7) is also presented:

Piece	1.02.01.02	
Exerted force (N)	F= 0,316	
Force diagram	Represented in study images	
Failure criteria	VonMises	
Mechanical test Min Max	**Min**	**Max**
Stress (N/mm^2)	$3,67 \cdot 10^{-4}$	2,449
Displacement (mm)	$1 \cdot 10^{-30}$	$3,801 \cdot 10^{-2}$
Strain	$1,832 \cdot 10^{-7}$	$7,245 \cdot 10^{-4}$
Factor of Safety (FOS)	3	-

Table 3.7 Modified Crown gear's static study results. Source: Own

Considering the obtained results (Table 3.7), and more specifically the Factor of Safety (FOS) (Figure 3.19d), it can be concluded that the modified crown gear piece can fulfill the mechanic requirements while keeping the imposed teeth geometrical restrictions.

- **Tests of norm-accurate figurative gear mechanism:** According to the gear-teeth normalization and regarding the previously designed gear-teeth trapezoidal geometry (Figure 3.4), it must be commented that the norm inaccuracy of the already justified pieces has been modeled to ease the design and the optimization process of the gear-based mechanism, given the freedom in terms of dimensions that this trapezoidal geometry can offer. However, this mechanism is not meant to be a final solution to apply in real nanosatellite launching conditions.

  Consequently, and so as to give an easily repeatable norm-accurate gear-based deploying mechanism for origami-based solar panel typologies, the following alternative gear assembly has been designed by using the ANSI Metric toolBox available in SolidWorks mechanical CAD software (Figure 3.20):

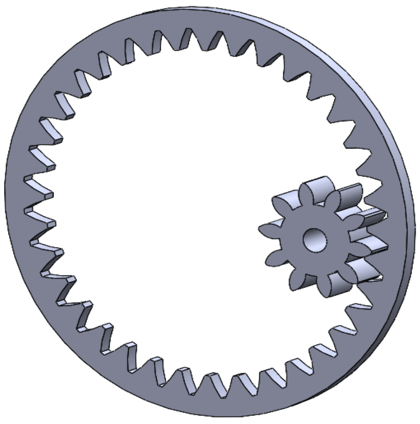


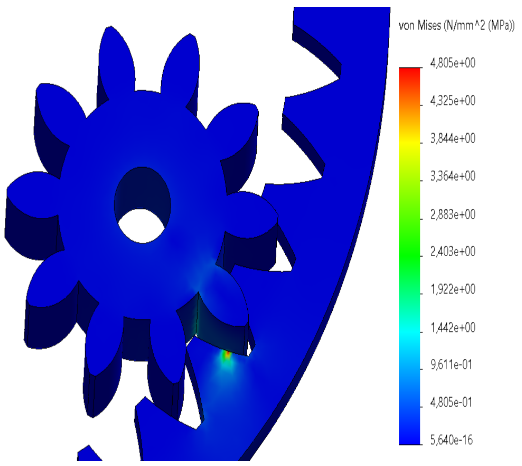
Figure 3.20 Norm-accurate gear mechanism figurative 3D model. Source: Own

The fundamental dimensions of this figurative gear mechanism are (Table 3.8):

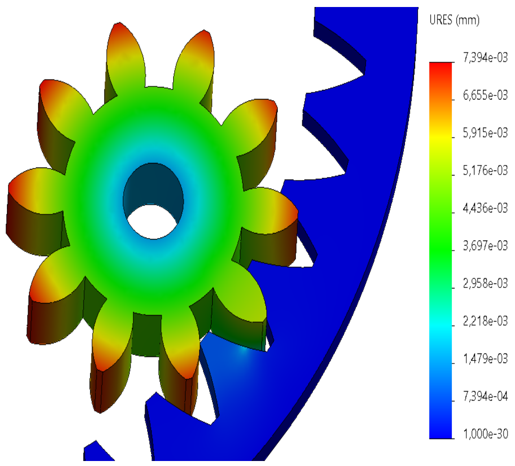
	Crown gear	**Pinion gear**
Outside diameter, Do (mm)	68	24
Root diameter (mm)	77	15
Whole depth, ht (mm)	4,5	4,5
Circular pitch, p (mm)	6,34	6,258
Module or modulus, m	2	2
Number of teeth, N	36	8
Reduction ratio	4,5/1	

Table 3.8 Fundamental dimensions of figurative gear mechanism. Source: Own

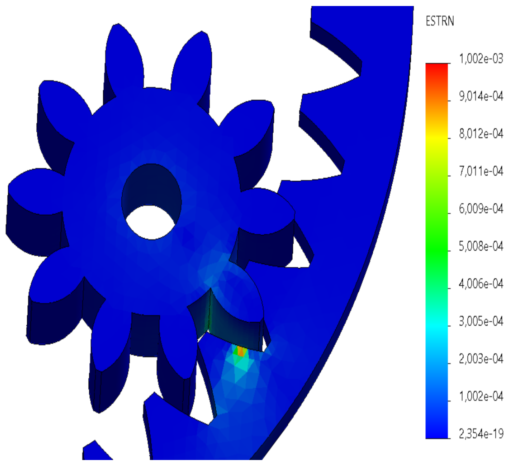
According to the given geometry and to ensure its technical viability, a Finite element modeling simulation has been performed, resulting in the following graphs (Figure 3.21):



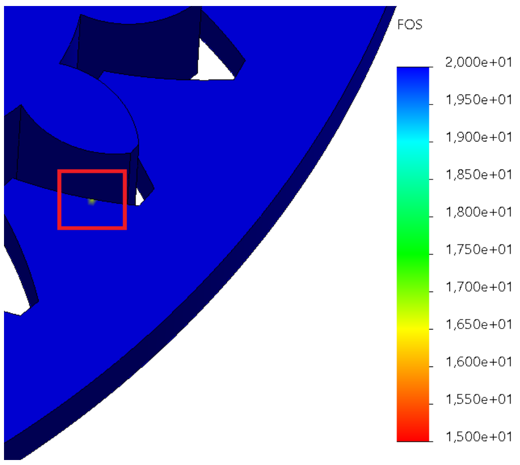
(a) Static study's Stress diagram.



(b) Static study's Displacement diagram.



(c) Static study's Strain diagram.



(d) Static study's Factor Of Safety diagram.

Figure 3.21 Norm-accurate gear mechanism's static study diagrams. Source: Own

***Note:** The red square corresponding to the FOS simulation has been placed to remark the color variation in the figure.

Given the previously seen Finite element modeling diagrams, the following results can be extracted and analyzed (Table 3.9):

Piece	Figurative gear mechanism	
Exerted torque (mN· m)	M= 34,3	
Force diagram	The torque is exerted over the pinon-gear	
Failure criteria	VonMises	
Mechanical test	**Min**	**Max**
Stress (N/mm^2)	$5,547 \cdot 10^{-16}$	4,904
Displacement (mm)	$1 \cdot 10^{-30}$	$7,906 \cdot 10^{-3}$
Strain	$1,556 \cdot 10^{-19}$	$1,093 \cdot 10^{-3}$
Factor of Safety (FOS)	14	-

Table 3.9 Norm-accurate gear mechanism's static study results. Source: Own

Considering these results, more specifically the FOS result (Figure 3.21d), it can be concluded that this figurative approximation to a norm-accurate gear mechanism could fulfill the required mechanical limitations for the proposed solar panel deployment system even in academic test conditions.

3.2.2.1.2 Revision of the dimensional tolerances:

Some of the malfunctions that have been identified in the first prototyping iteration can be also solved by revising and adjusting the dimensional tolerances applied to the junctions between some pieces, such as:

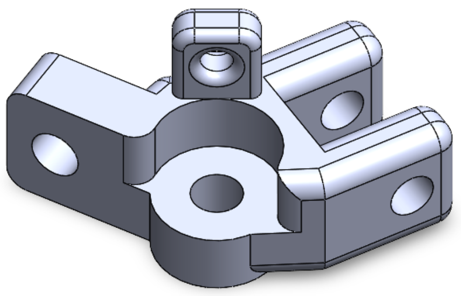
- **Rhomboidal deployable arms:**
  During the testing of the first prototyping iteration, it has been observed that the rigidity of the junctions between the deployable arm rods is affected when the deployment maneuver is repeated a great number of times. This deterioration of the rods can be solved, or at least retarded, by adjusting the dimensional tolerance of these pieces in order to stiffen these junctions while allowing the free rotation of the designed pieces.
- **Gear mechanism:**
  Once the modifications over the gear mechanism have been redesigned, it has been detected that the force exerted by the pinion gear against the crown gear, due to the modification of the dimensions of these parts, has caused a movement of the drive shaft of the stepper motor, thus making its movement impossible.
  In order to solve this problem, it is necessary to modify the dimensional tolerances between these toothed parts so that, although allowing a good fit between the gears, does not affect the movement of the motor shaft.

Regarding the previously mentioned observations, it must be taken into account that, because of the characteristics of the used material and the fact that the studied pieces have been manufactured by using 3D printing technologies, the adjustment of the specified dimensional tolerances has been achieved through the use of precision drill bits after the manufacture of the pieces.

3.2.2.2 V-shape rod structure

Once the deployment mechanism has been completely developed and implemented and with the purpose to also implement an improvement over the solar panel unfolding mechanism, the concept of adding a V-shape structure on the tip of the deployable arms was studied as a consequence of a more exhaustive observation of the NASA Starshade program [8].

Therefore, to give a case specific solution to this concept, the following parts were designed (Figure 3.22):


(a) V-shape structure hinge 3D model.

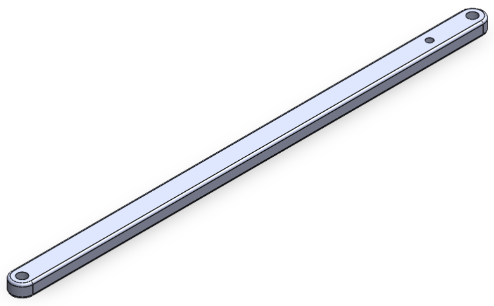
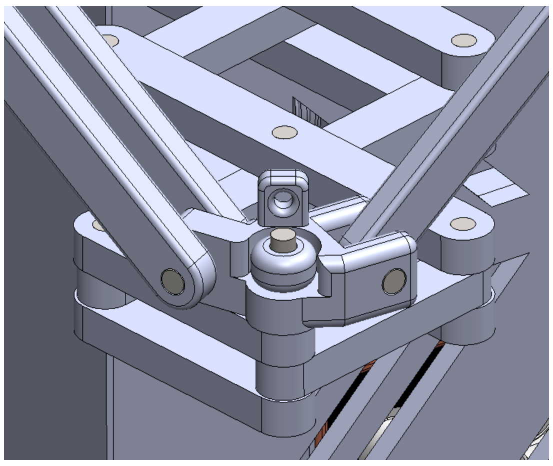

(b) V-shape structure rod 3D model.

Figure 3.22 V-shape structure fundamental parts 3D models. Source: Own

To design the already shown parts, the determination of the length of the V-shaped rods (Figure 3.22b) is not a trifling matter, since this longitude will condition the unfolding of the whole structure. Aiming to perform a proper sizing of these pieces, the dimensional characteristics of the used origami pattern has been taken into account (Figure 3.15).

After the designing of new parts, the assembly to the final deployment prototype will be performed as follows (Figure 3.23):


(a) V-shape hinge assembly front view.

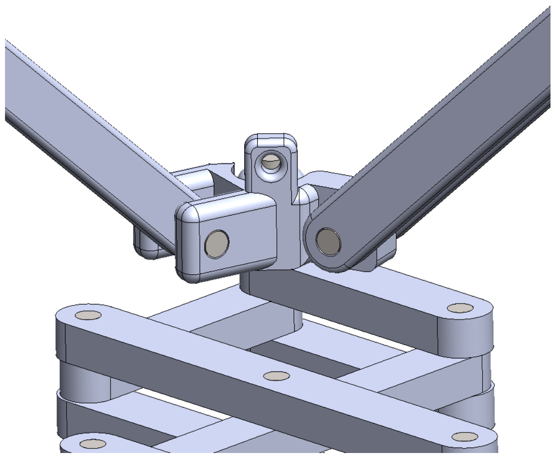

(b) V-shape hinge assembly back view.

Figure 3.23 V-shape hinge assembly 3D model view. Source: Own

Finally, by attaching the origami pattern to this V-shaped structure using wires, the completely assembled and functional origami-based solar panel deployment system prototype is ready to be tested (Figure 3.24).

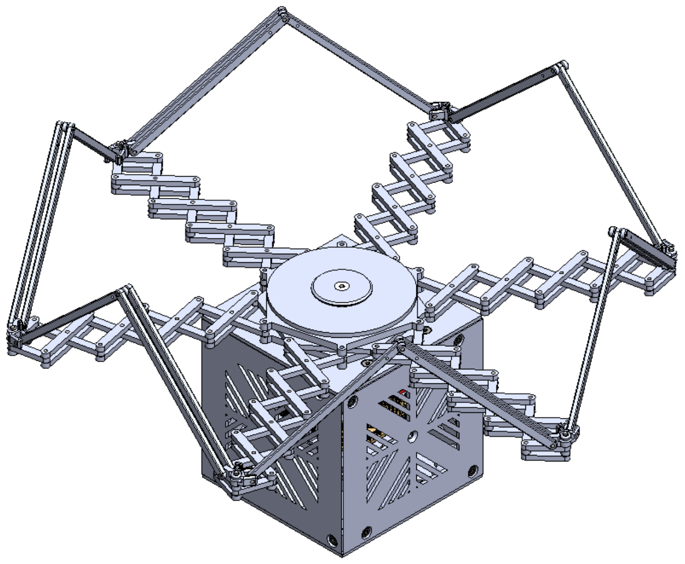


Figure 3.24 Final prototype assembly 3D model isometric view. Source: Own

Chapter 4

Electronics

4.1 Introduction to the ADCS code

This chapter of the project will be dedicated to the introduction, explanation, and justification of the electronic modifications and improvements over the previous ADCS system, developed for Adrià Pérez and Yi Qiang Ji Zhang. Considering so, the first step of this section should be to provide an introductory explanation of the ADCS code, to ensure proper comprehension of the used devices and their operation.

To begin with, it must be mentioned that the implemented software code runs on the STM32 Bluepill microcontroller and is coded in Arduino programming language. When talking about the insights of the ADCS code, two different control algorithms have to be independently studied, which are:

- **Coarse positioning mode:** This operating mode is responsible of the first approximation to the desired position and it covers most of the angular distance and maneuver time. This movement is performed by applying an "uncontrolled" voltage pulse over the DC motor that actuates over the reaction wheel, which will be later stopped when the spacecraft is positioned whitin the Fine-pointing tolerance region.
- **Fine positioning mode:** Once the spacecraft is close enough to the desired orientation, thanks to the coarse pointing mode operation, the fine positioning mode is turned on to exert a more accurate control of the final movements. Moreover, the positioning maneuver executed by this mode can be controlled by using the two following control algorithms, which will be better explained later in their respective sections.
  - Simple control algorithm, executed by an accurately tuned PID controller whose constants are determined by applying empiric methods.
  - Precise control algorithm, executed by a more complex control loop which also controls both electric and mechanical factors like current or applied torque over the reaction wheel.

The following picture plots the position of the satellite over the entire maneuver, therefore showing the operation of both positioning modes (Figure 4.1):

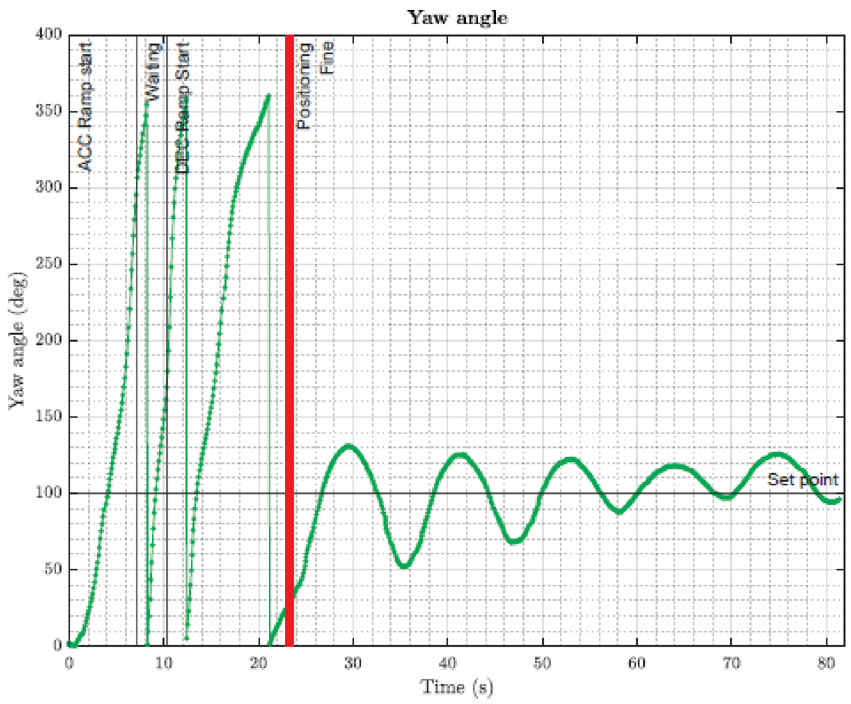


Figure 4.1 Positioning modes identification graphic representation. Source: [1]

4.2 Problems to face

When the results of previous ADCS-related projects were replicated [1] [2], it was found an inaccuracy in the attitude control maneuvers, which was caused by a series of independent problems affecting both of the previously introduced pointing modes. In this section these problems will be properly explained and justified and, at the same time, the proposed solutions will be exposed and argued in order to ensure an easy and reliable comprehension of the explained testing procedures.

In order to do so, each one of the after mentioned problems will be analyzed and solved following the three steps presented below:

- Identification of the problem: First of all, and aiming to obtain a properly implemented and studied solution, the problem that produces the malfunction of the device during the execution of its task must be precisely identified.
- Further explanation of the identified problems: Before the exhaustive explanation of the solutions that have been developed to achieve the optimum behavior of this device, and aiming for a proper understanding of the readers of this report, the explanation of the previously identified problem must be further extended.
- Possible solutions researched for these problems: Once the problem is properly introduced, a detailed explanation of the studied solutions will be carried out to ensure a proper understanding from the readers.

4.2.1 IMU measurement problems

During the testing of the IMU's performance at the measurement of the Euler angles variations over rotation, it was observed a meaningful accuracy issue at the angular position measurement, which produced and incorrect movement of the device. Considering so, it is found that the original Euler angle calculations were performed by using the direct measurement from the magnetometer, accelerometer, and gyroscope sensors of the Inertial Measurement Unit (IMU), which led to the previously mentioned accuracy problems due to the following facts:

- The use of direct magnetometer measurements in the positioning calculation proved to be excessively unreliable due to distortions on the surrounding magnetic field produced by other devices in the test-filed, such as mobile phones, batteries or other devices in the lab.
- The use of direct accelerometer measurements produces a residual noise in the measurement due to the vibration of the experimental equipment used over the testing process, which leads to a distorted measurement of the final position of the device. To exemplify this affirmation, see the following image (Figure 4.2):

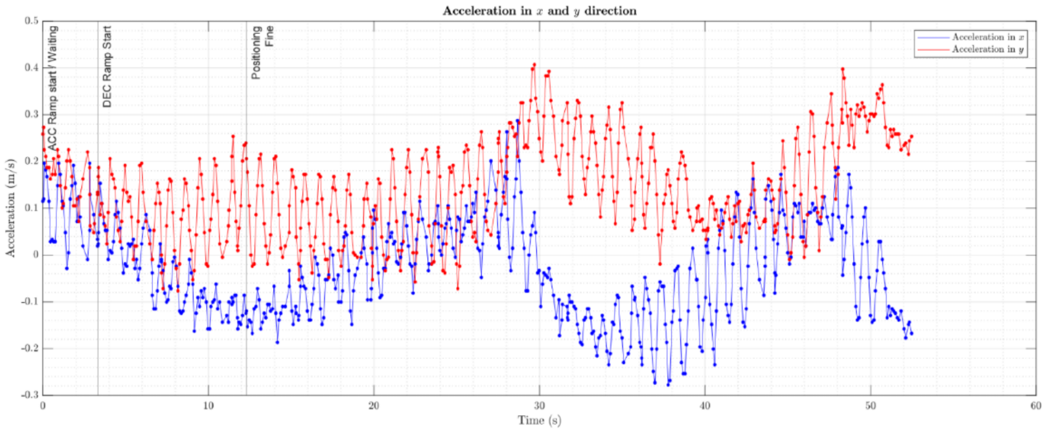


Figure 4.2 Acceleration noise graphic representation. Source: [2]

However, the previously exposed distortion can be overcome with the implementation of some filters to enhance the accuracy of these measurements.

Considering the previous problems, and aiming to achieve a completely accurate measurement of the orientation of the CubeSat, a DMP (Digital Motion Processor) device, already supplied by the MPU9250 module, has been used to process the recorded IMU data and obtain the desired orientation in form of quaternion equations.

Once the previously mentioned quaternions are obtained the calculation of the Euler's angles was possible by using the software Arduino library 'MPU6050_6Axis_MotionApps20.h' and the subroutines named 'IMU()' and 'QuatToEuler()' developed by Juan Palomares Moyano [30] (while IMU() order executes the subroutines used to perform the DMP access, named 'mpu.dmpGetQuaternion', 'QuatToEuler()' calculate the Euler's angles by using the quaternion equations obtained from the DPM).

Taking into account the previously performed superficial explanation of the software process followed to obtain the orientation angles, it is time to explain the hardware characteristics and changes that were implemented so as to obtain a functional solution to the measurement problems.

4.2.1.1 Digital motion processor (DMP)

Since it has been already mentioned in the previous section, and due to the fact that it is one of the most relevant systems in the improvements that have been implemented in the final CubeSat prototype, a deeper explanation of the Digital Motion Processor (DMP) is required.

Considering so, and remembering that the MPU-9250 module containing the IMU also presents an internal DMP, it is found that the main function of this type of processor is to combine the measurements taken from the gyroscope, accelerometer, and magnetometer devices. Therefore, the presence of the DMP allows to obtain an already filtered output derived from the raw experimental data, which implies that an external filter application is not necessary, consequently reducing the calculation time and the microcontroller memory requirements. Due to these advantages, and given that the results obtained by using the DMP integrated device are more accurate than by applying a complementary filter, the attitude control position data has been calculated by using the Digital Motion Processor.

Despite this great improvement in terms of the measurement process, the DMP also produces side disadvantages derived from the lack of information about the operation process of this device, since, due to industrial secrecy of the manufacturer, it is found impossible to access, modify or replicate the calculation process performed by the DMP.

Once this introduction to the DMP technology has been explained, a deeper explanation of the DMP output data must also be commented.

The data previously processed by the DMP can be then obtained by using two different approaches, as an attitude quaternion equation or as an Euler's angle. Taking this into account, and given the inaccuracies observed at Euler's angles due to Gimbal lock [31], the final output data will be extracted as attitude quaternion equations, which also implies the following advantages for its processing:

- The mathematical conversion from quaternion equations to Euler's angles is easier than the reverse one.
- The gimbal lock can be better controlled by using quaternion equations for position data processing. (Even though the gimbal lock phenomenon does not occur when performing a single-axis attitude control, the possibility to modulate this effects can be profitable when implementing multiple-axis attitude control.)

The following image 4.3 exemplifies the block diagram of MPU-9250, also including the DMP device interconnection:

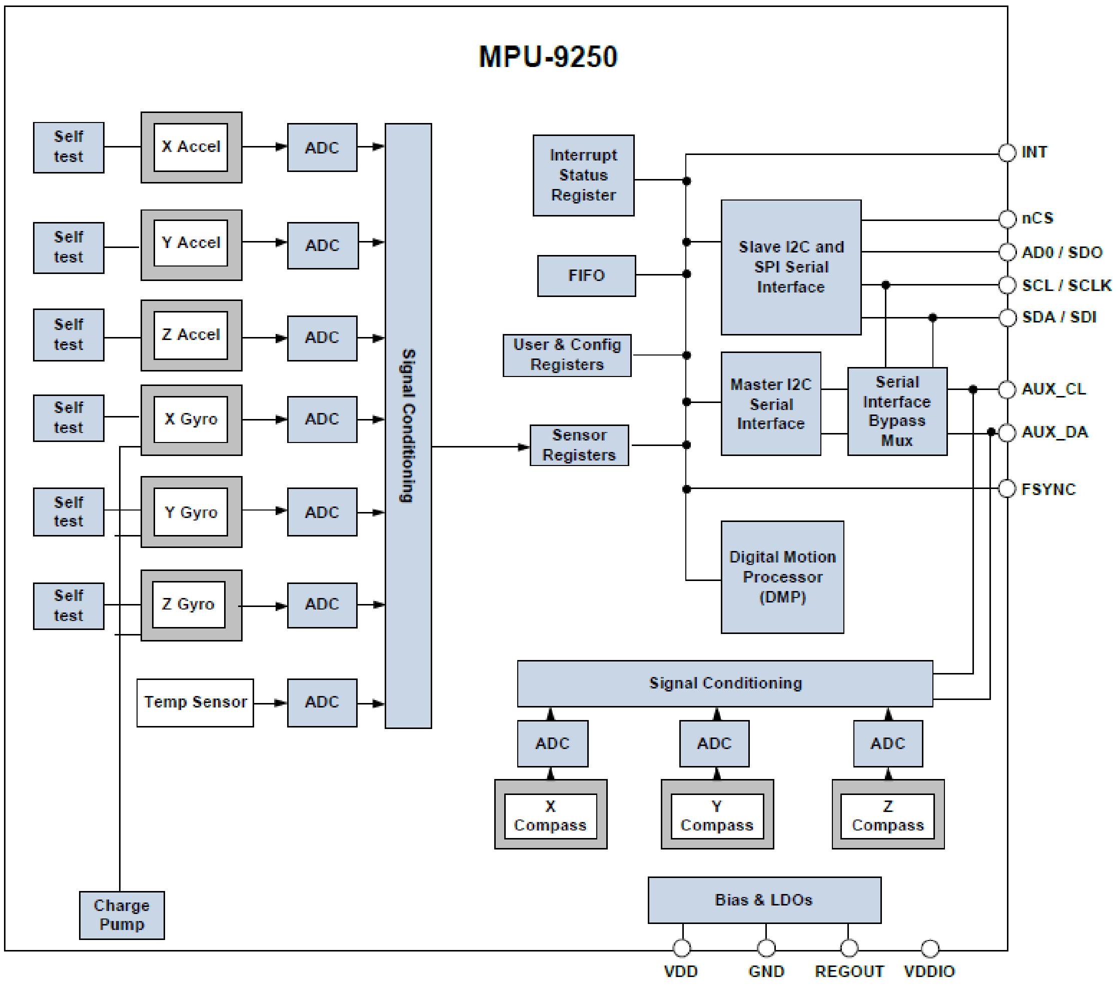


Figure 4.3 MPU-9250 block diagram. Source: [32]

4.2.1.2 Communication bus modifications

As the project goes on, it is of paramount importance to explain the changes in terms of communication buses that had to be implemented to obtain a proper data transmission between the different devices on the CubeSat, including the IMU and DMP previously discussed.

Originally, the communication between the MPU device and the microcontroller in charge of the data acquisition and processing was made by using a serial peripheral interface (SPI) communication bus, while the communication between the motor driver and the same microcontroller was performed using an I^2C communication bus.

For more information, see the image below (Figure 4.4):

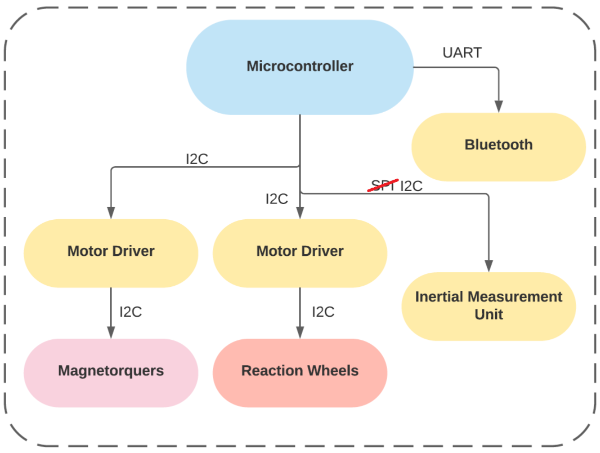


Figure 4.4 Used communication buses scheme diagram. Source: Own

Considering the communication previously described, it was discovered that, because of the communication specifications of the libraries that were used to access the DMP device (the 'MPU6050_6Axis _MotionApps20.h' library), the only possible way to connect the MPU to the microcontroller while still allowing access to the DMP was by using I^2C bus communication.

The first attempt to change the communication connections tried to take advantage of the multiple I^2C communication ports that the used microcontroller board (the STM32 Bluepill model) offers, to use two independent I^2C units to connect the microcontroller with the MPU and the motor driver without sharing the same bus.

This approach was attempted due to the reported identification problems that can arise when both devices are connected through the same bus, since serious malfunctions can occur in this case if the address assignment of the final code is not completely accurate. However, performance tests of the double I^2C system showed that they did not perform good data transmission between the devices, and the same I^2C bus was finally used for both MPU and motor driver connection, always ensuring the use of different addresses for every device access.

After considering these decisions, the following test was performed in order to ensure proper communication and measurement of the MPU device, despite that the DC motor driver connection couldn't be tested by the presented setup given it is only present at the hardware PCB (Figure 4.5).

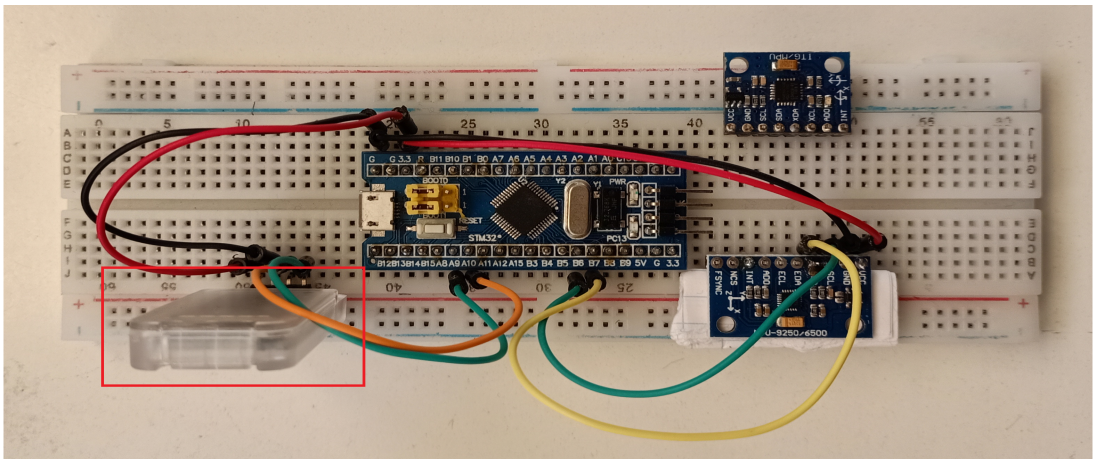


Figure 4.5 Protoboard real implementation of IMU communication circuit. Source: Own

The previous test had the main objective of ensuring the correct function of the proposed connections between the different devices used for the positioning measurements, as well as checking the proper data acquisition and processing of the Euler's angles during the device movement, produced by rotation of the protoboard.

In order to precisely specify the connections performed in the previous image, the following scheme 4.6 must be shown (where all performed connections in the final PCB hardware are specified):

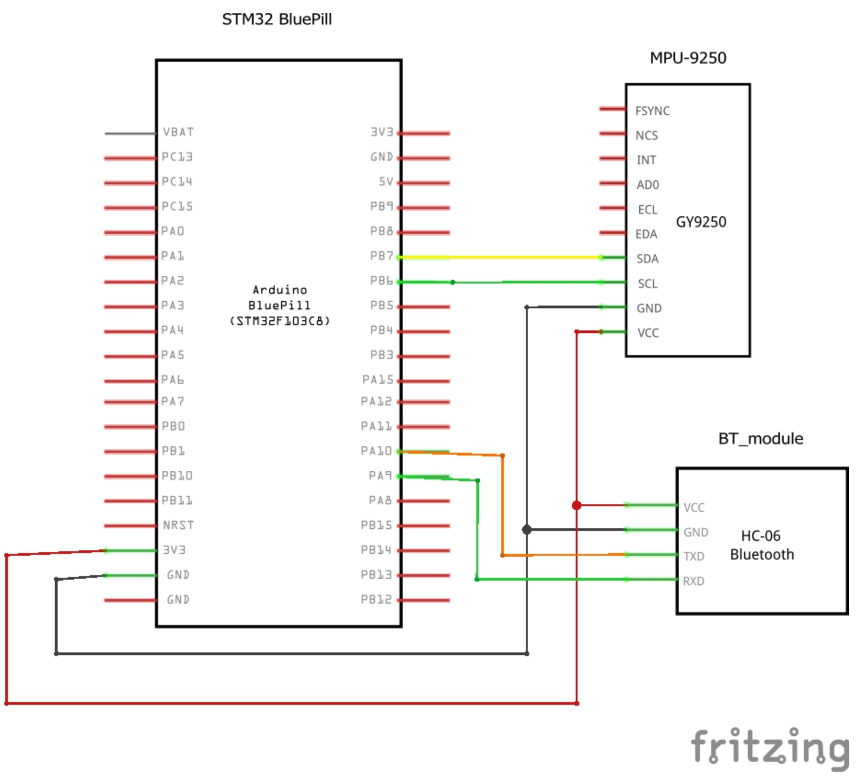


Figure 4.6 Electric scheme of IMU communication circuit. Source: Own

The following table 4.1 is used to illustrate the after mentioned connections:

MPU pin	**HC-06 Bluetooth pin**	**Bluepill pin**	**Color Code**
SDA (I^2C data bus)	-	B7 pinout	Yellow
SCL (I^2C clock bus)	-	B6 pinout	Green
-	TDX	A10 pinout	Orange
-	RXD	A9 pinout	Green
VCC	VCC	3v3	Red
GND	GND	GND	Black

Table 4.1 IMU communication circuit's connections color code. Source: Own

By using the previous graphically exemplified test (Figure 4.5) it has been achieved proper communication of the MPU, by using the DMP access and the I^2C communication bus at the same time, as well as also ensuring the MPU communication with the OBC (On-Board Computer) by using the Bluetooth module (highlighted in red in the previous protoboard image).

Once the previous test has been executed, the modifications on the hardware PCB board to change the original SPI communication into an I^2C communication are completed, as can be seen in the following image (Figure 4.7):

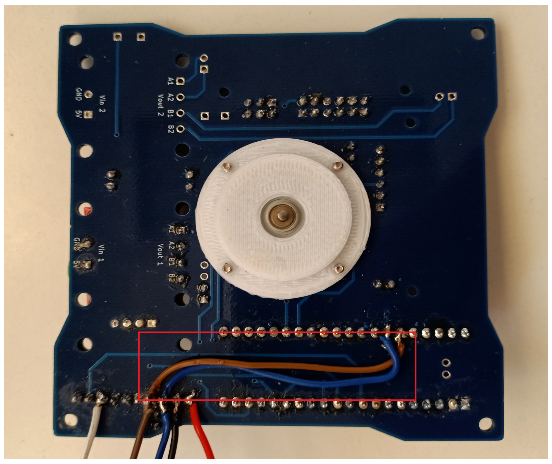


Figure 4.7 IMU communication circuit modifications over the prototype's PCB. Source: Own

The pin interconnection between the MPU and the STM32 Bluepill board shown in the previous image (Figure 4.7) was performed using the following color code (Table 4.2):

MPU pin	**Bluepill pin**	**Color Code**
SDA (I^2C data bus)	B7 pinout	Brown
SCL (I^2C clock bus)	B6 pinout	Blue
GND	Included in the PCB design	Black
3v3	Included in the PCB design	Red

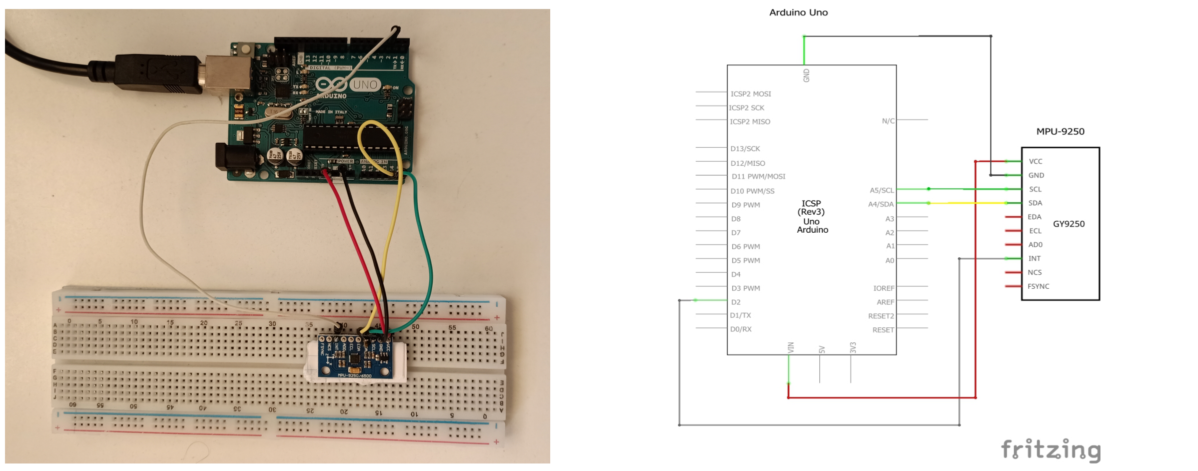
Table 4.2 PCB's IMU communication circuit color code. Source: Own

4.2.1.3 Calibration process modifications

While performing the previously explained test (Figure 4.5) it was found that, more than just performing a proper communication and access to the DMP, a proper calibration of the MPU device, considering the external conditions existing in the lab, must also be performed to achieve a correct measurement of the MPU. With this objective in mind, the following tests were designed.

First of all, it must be mentioned that the program used in order to obtain the offset constants used to calibrate the MPU was the named 'MPU6050 calibration.ino' (See Appendix D.2), and so as to execute it, the Arduino UNO board microcontroller was used, because of some problems with code execution in a different board.

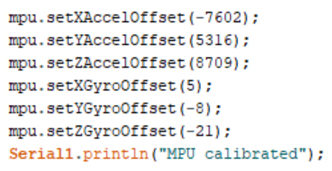
Having this fact in consideration, and aiming to do a proper calibration, a preliminary test by using a protoboard was performed (as in the case of the measurement tests), as it can be seen in the following image (Figure 4.8).



(a) MPU's calibration real protoboard test. **(b)** Electric scheme of MPU calibration protoboard test.)

Figure 4.8 MPU's calibration protoboard test. Source: Own

All these tests (Figure 4.8) were performed with the main objective to simplify the calibration process of the MPU, which was quite complex in the original code, while ensuring a good enough precision on the measurements after the calibration.
The process to do the calibration, having into account the connections that can be seen in the previous image (Figure 4.8b), is performed by positioning the MPU on a flat surface (which can be, for example, a table or even the floor) while executing the calibration code already specified. After waiting some seconds, the offset constants can be obtained by using the Arduino serial port as a communication bus (Figure 4.9).

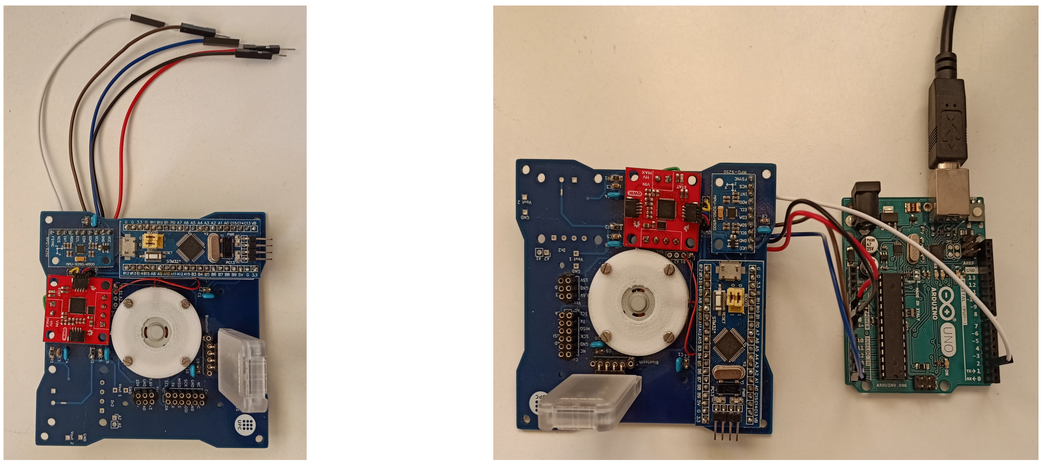


```
mpu.setXAccelOffset(-7602);
mpu.setYAccelOffset(5316);
mpu.setZAccelOffset(8709);
mpu.setXGyroOffset(5);
mpu.setYGyroOffset(-8);
mpu.setZGyroOffset(-21);
Serial1.println("MPU calibrated");
```

Figure 4.9 MPU's real Calibration offsets. Source: Own

Once the proper behavior of the calibrated MPU is completely ensured, and after performing the previously described test (Figure 4.8), the implementation of the same connections on the PCB hardware can be finally performed (Figure 4.10).

The followed procedure, aimed to obtain the final offsets that will be used in the final PCB hardware device, is the same as the one that has been already explained. The main difference is that, in this case, it has been considered that the motor driver is also connected to the microcontroller by using the same I^2C communication bus, which can be easily solved by switching off the motor driver during the calibration process.



(a) PCB's calibration wires. **(b)** PCB's calibration final connections.

Figure 4.10 Final MPU's calibration on PCB. Source: Own

The color code used to identify the connections between the MPU device and the Arduino UNO board shown in the previous image is the following one (Table 4.3):

MPU pin	**Bluepill pin**	**Color Code**
SDA (I^2C data bus)	A4 analog input	Brown
SCL (I^2C clock bus)	A5 analog input	Blue
GND	GND	Black
Vcc	5v	Red
INT (external interruption pin)	Digital (PWM) 2	White

Table 4.3 MPU's calibration connection's color code. Source: Own

4.2.2 Coarse Mode: Overrotation due to large ramps

The main problem that has been identified during the execution of the coarse positioning mode is the overrotation (i.e. excessive rotation) of the device due to the large voltage ramps applied to the motor. The behavior of the device can be seen in the following picture 4.11:

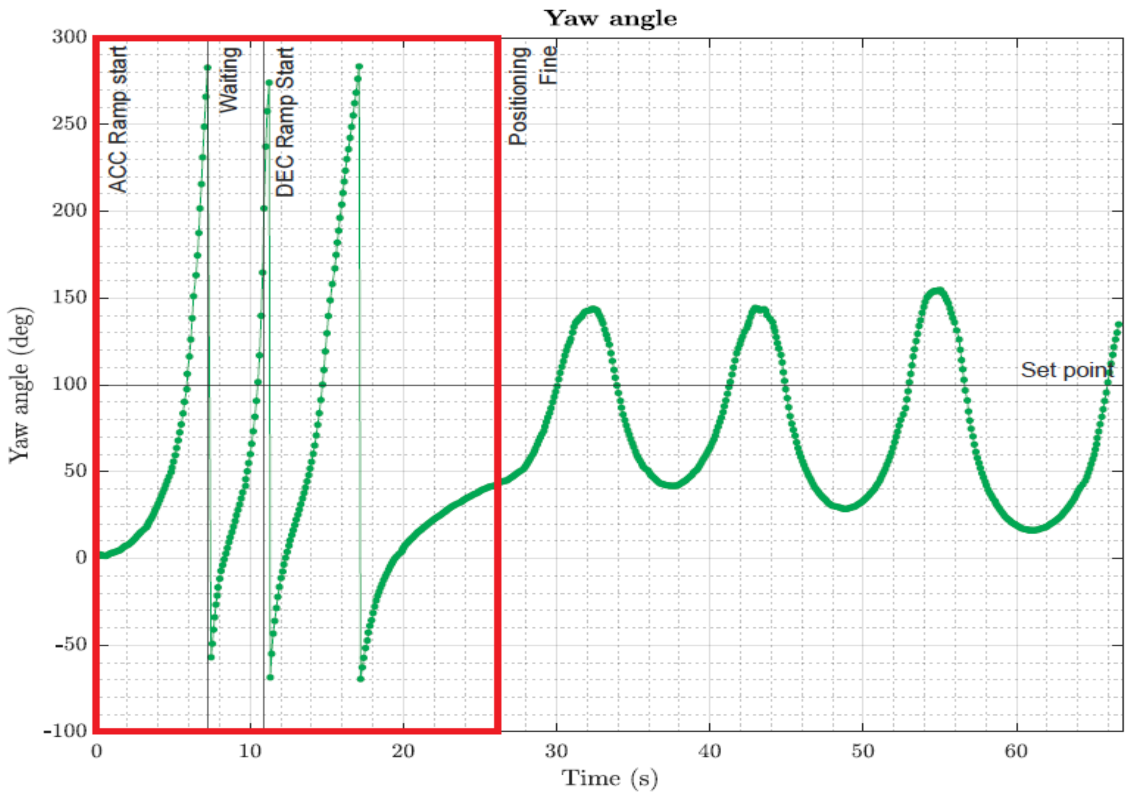


Figure 4.11 Graphical representation of overrotation in Coarse pointing mode. Source: Own

Here can be seen how, in the coarse phase, the prototype completely rotates 3 times before entering the fine mode, which becomes a meaningful issue due to the low efficiency in terms of power consumption and operation time.

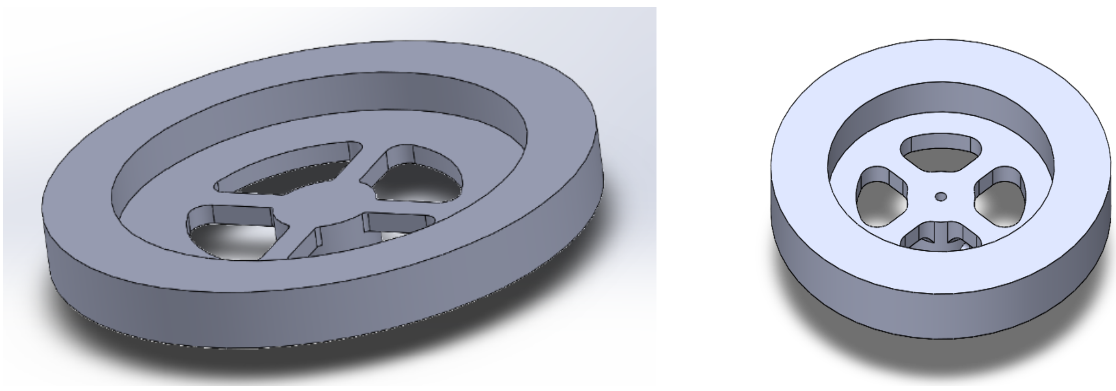
Once the problem to solve has been identified, it has been decided to study the possible solutions focusing on two different branches of knowledge. These possible solutions were studied in terms of both, the cinematic properties of the reaction wheel (also named RW in this written report) used to perform the maneuver of the device, and the software modifications of RW control system used to change its behavior during this positioning mode.

The following parts of this report will present an in-detail description of the developed solution, classifying them according to whether they are hardware-based or software-based improvements.

4.2.2.1 Hardware focused solution: Reaction wheel redesign

Since this chapter is directed to the development and justification of the electronically-related parts of this project, the hardware solutions will be better explained in Chapter 5 (see Inertia calculation section). However, a brief explanation of the solution can be found below.

The main objective of this hardware-focused solution is to avoid the overrotation of the spacecraft during the Coarse pointing mode maneuver. To do that, a reduction of the inertial momentum of the reaction wheel in charge of moving the spacecraft will be performed.



(a) Reaction Wheel's original design. (b) Reaction Wheel's improved design.

Figure 4.12 Original/Modified Reaction Wheel's design. Source: Own

The observed size reduction between Figure 4.12a and Figure 4.12b will produce a reduction of the momentum of inertia of the system, and therefore a slower and smoother movement of the hole device during the Coarse maneuver.

All these considerations and deductions will be mathematically justified in Section 5.3.

4.2.2.2 Software focused solution: Coarse pointing program modification

As it has been already mentioned, the Coarse positioning mode is the main responsible of the control of the largest movement that the device will experience during the maneuver commanded by the operator. Taking into consideration that this movement does not require a great degree of precision, the main objective of this positioning mode is to get the device as close as possible to the set point prefixed by the external operator, and sent to the device by using the Bluetooth module.
Once the main function of this positioning mode has been properly identified and considering the previously stated overrotation problem we will proceed to the explanation of the modifications performed in the program for the purpose of solving this problem.

Aiming to provide a proper background to this process, regarding the previous work done by Yi Qiang Ji Zhang [1] and Adrià Pérez Fernández [2], the function of two of the subroutines used in the original program that affect the maneuver (more specifically the Coarse maneuver) will be briefly explained:

- **generate_ramp**: The original function of this subroutine was to allow the operator to properly control the acceleration phase after setting the desired speed. This attempt to control the acceleration of the device was meant to also control the forces acting over the spacecraft by modifying the acceleration ramp used to accelerate the RW.
  However, this subroutine was the main responsible of the problems faced during the first tests of the original program, and it was consequently eliminated from the final control program.

- **set_impulse**: The main function of this subroutine is to send a speed set point to the motor driver, by using a PWM (pulse width modulation) signal, as a way to modify the rotation speed of the reaction wheel and, consequently, the rotation speed of the CubeSat. By using this subroutine, it can also be specified if the rotation will be clockwise or counterclockwise.

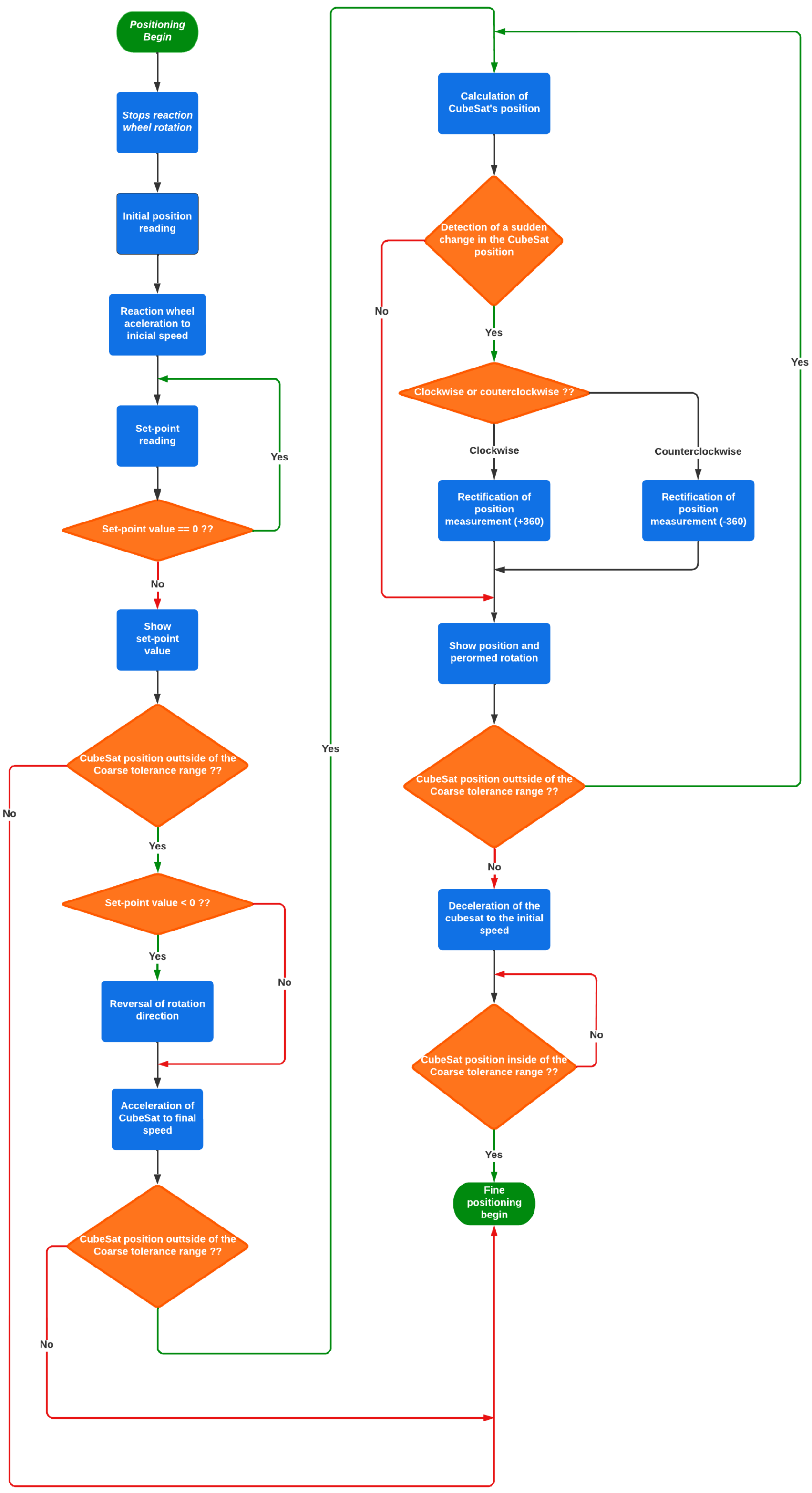

Positioning Begin
Stops reaction wheel rotation
Initial position reading
Reaction wheel aceleration to inicial speed
Set-point reading
Yes
Set-point value == 0 ??
No
Show set-point value
CubeSat position outtside of the Coarse tolerance range ??
No
Yes
Set-point value < 0 ??
Yes
No
Reversal of rotation direction
Acceleration of CubeSat to final speed
CubeSat position outtside of the Coarse tolerance range ??
No
Yes
Calculation of CubeSat's position
Detection of a sudden change in the CubeSat position
No
Yes
Clockwise or couterclockwise ??
Clockwise
Counterclockwise
Rectification of position measurement (+360)
Rectification of position measurement (-360)
Show position and peromed rotation
CubeSat position outtside of the Coarse tolerance range ??
Yes
No
Deceleration of the cubesat to the initial speed
CubeSat position inside of the Coarse tolerance range ??
No
Yes
Fine positioning begin

Figure 4.13 Coarse Pointing Mode flowchart (Augmented version in Appendix D.1). Source: Own

4.2.3 Fine Mode: System instability due to deficient implementation of the control algorithm

Once the previous Coarse-Pointing solutions have been explained and implemented, the research of some solutions for the Fine-Pointing mode accuracy improvement must be considered. So as to ensure proper precision in the final stage of the positioning maneuver, an optimum attitude control algorithm must be a fundamental requirement. To achieve this goal, two different approaches (in terms of the control loop and experimental procedure) have been studied:

4.2.3.1 PID controller tuning by an empirical method:

In order to achieve maximum simplification of the mathematical calculations involved with the tuning process, a simple single-controller PID control loop has been studied. The proposed closed control loop is the following (Figure 4.14):

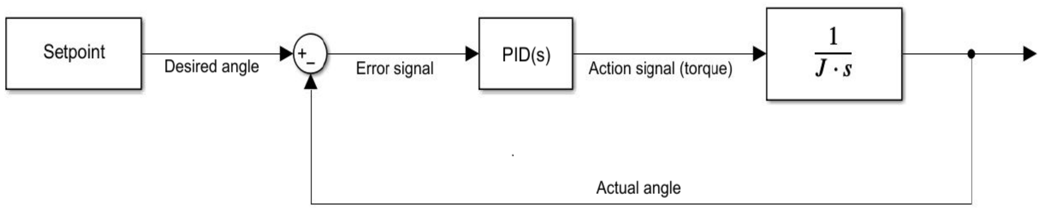


Figure 4.14 PID control loop block diagram. Source: [33]

To give a proper introduction to PID controllers, the following mathematical expression of the control signal must be considered (Equation 4.1):

$$u(t) = \underbrace{K_p e(t)}_{Proportional} + \underbrace{K_i \int e(t)dt}_{Integral} + \underbrace{K_d \frac{de}{dt}}_{Derivative} \tag{4.1}$$

where:

Variable	**Description**
u(t)	Resulting control signal
e(t)	Error signal of controlled variable
$\mathbf{K_p}$	Proportional gain
$\mathbf{K_i}$	Integral gain
$\mathbf{K_d}$	Derivative gain
de	Differential of error
dt	Differential of time

Table 4.4 Descriptive table of PID's mathematical expression. Source: [33]

As it can be seen in the previous expression 4.1, a PID control response can be divided into three different contributions, which are:

- **Proportional contribution:** This contribution is based on the correction of the error proportionally to the Kp value. This constant is closely related to the response speed of the whole system. However, as the error approaches zero, so does the correction action, meaning that it may never reach the set-point specific value.
- **Integral contribution:** The Ki constant is the most closely related to the steady-state error value since it fixes the final difference between the set-point and the actual measurement, solving the proportional action problem. Regarding the fact that, in spacecraft application, this contribution is not considered, Ki will be set to 0 by definition in the final proposed solution.
- **Derivative contribution:** Derivative contribution directly acts over the rate of approximation of the controlled variable (in this case, the position value) to the set-point value. This implies that it minimizes the overshoots in the error signal that may appear due to the corrections of the proportional and integral actions.

This PID-based control loop has the main advantage of its simplicity, but, according to the limitations in terms of the real plant modeling possibilities, it also has an extremely restrictive disadvantage, which is the fact that the only possible way to determine the control constants is by applying empirical methods.

The tuning process to achieve a good enough control of the whole system would be based on the determination of preliminary constants by using empirical approximations such as, for example, the Ziegler-Nichols empirical method.

Control	**P**	**Ti**	**Td**
P	0,5·$K_{p,cr}$	-	-
PI	$0,45 \cdot K_{p,cr}$	$T_{p,cr}/1{,}2$	-
PID	0,60·$K_{p,cr}$	0,5·$T_{p,cr}$	$T_{p,cr}/8$

Table 4.5 Ziegler-Nichols empirical constants determination table. Source: Own

This method is based on a trial and error concept. First, both integral and derivative constants (Ki and Kd) are set to 0. After that, Kp is tuned to achieve an oscillating and slightly overshooting response. Then, the critical constant Kp,cr, and period Tp,cr are used in order to determine a first approximation of the Kd and Ki values according to table 4.5. Finally, the original series of Kp, Kd and Ki are used for an iterative search of the optimal PID response.

4.2.3.2 Cascade control loop:

Since it will be very difficult for the previously seen PID control loop to properly control the behavior of the whole spacecraft, given the required great precision in the constant determination, a P+PI cascade control loop has been also studied. The basic concept of this P+PI method is the nesting of two PID algorithms feedback loops (Figure 4.15), but a detailed explanation of the control loop design process will not be performed here since it has been already detailed in Joan Serrano's Bachelor's thesis [33].

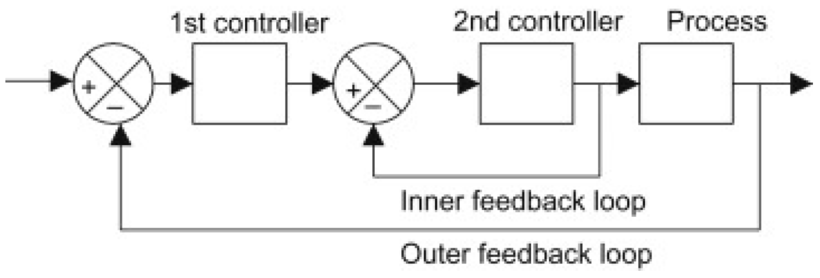


Figure 4.15 Cascade control loop block diagram. Source: [33]

As can be seen in the previous figure 4.15, the output of the first controller provides the set point of the second one. This loop structure can only be used when there is a dependency between the controlled signal of the inner loop and the controlled signal of the outer one, as shown in the following expression (Equation 4.2):

$$\int \alpha(t)dt = \omega(t) \longrightarrow \int \omega(t)dt = \vartheta(t) \tag{4.2}$$

The inner loop of this configuration is meant to control the angular speed of the spacecraft, which can be measured by using the designed prototype's hardware, with a PI controller, as shown in the figure (Figure 4.16).

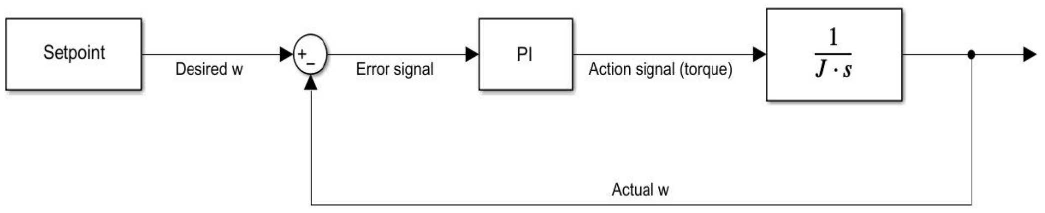


Figure 4.16 Angular speed inner feedback control loop block diagram. Source: [33]

Meanwhile, the outer loop, by using a Proportional controller, is in charge of controlling the final position of the CubeSat, as shown in the figure (Figure 4.17).

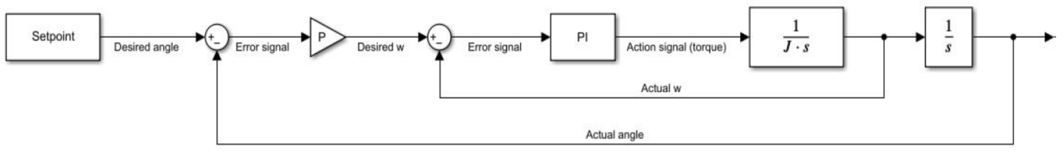


Figure 4.17 Position outer feedback control loop block diagram. Source: [33]

So as to determine the transfer function of the whole system and, consequently, the final control constants, the following attitude dynamics were fixed (Table 4.6):

Parameter	**Symbol**	**Value**
Setting time of position loop	$ST_{2\%p}$	10s
Setting time of angular speed loop	$ST_{2\%\omega}$	1s
Damping coefficient	ξ	1
Stationary gain	K	1
Momentum of inertia	J	$1,7 \cdot 10^{-3} (kg \cdot m^2)$*

Table 4.6 Attitude dynamics parameters. Source: [33]

***Note:** The momentum of inertia defined in the previous table is an estimation used by Joan Serrano for computational simulation testing purposes, the momentum of inertia corresponding to the developed prototype will be calculated in the following chapters.

The control constants will be referenced as follows (Table 4.7):

Parameter	**Symbol**
Proportional gain for the position controller	KP_P
Proportional gain for the angular speed controller	KP_W
Integral gain for the angular speed controller	KI_W

Table 4.7 Control constants' symbols table. Source: [33]

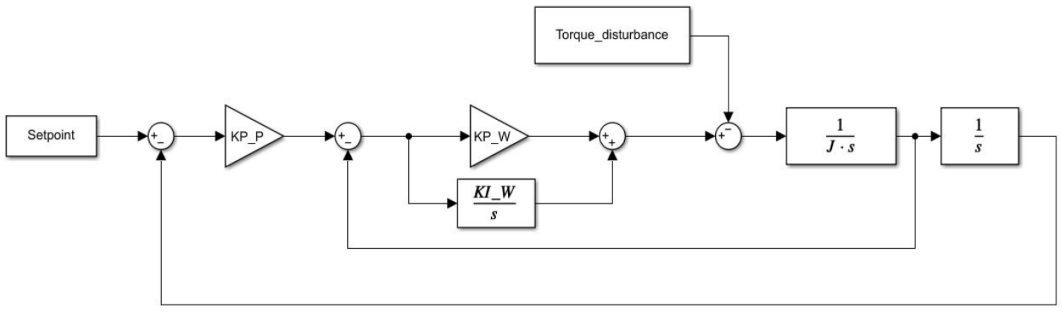


Figure 4.18 Final cascade control loop block diagram. Source: [33]

After implementing the following pre-filter (Figure 4.19), so as to avoid undesired zeros, the final Transfer Function of the system is proposed by Joan's project (Equation 4.3):

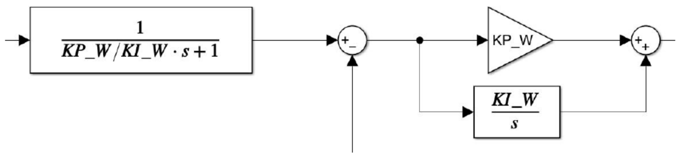


Figure 4.19 Pre-filter block diagram. Source: [33]

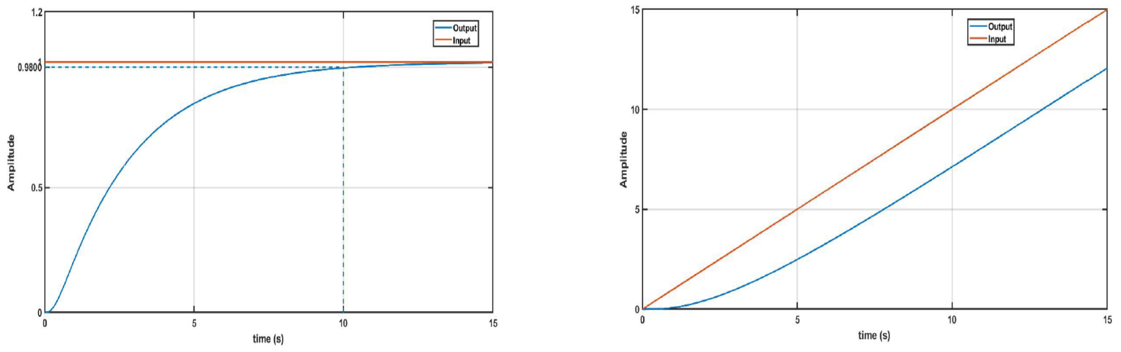
$$G(s) = \frac{n(s)}{d(s)} = \frac{KP_P \cdot \frac{KI_W}{J}}{s^3 + \frac{KP_W}{J}s^2 + \frac{KI_W}{J}s + KP_P\frac{KI_W}{J}} \tag{4.3}$$

And the resultant control constants, after applying the Root Locus procedure are obtained as (Table 4.8):

Constant	**Value**
KP_W	0,0193
KI_W	0,0561
KP_P	0,3402

Table 4.8 Root Locus method final Control constants values. Source: [33]

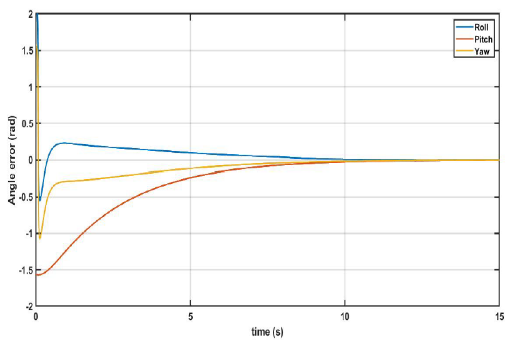
After some testing of the tuned cascade control loop, the following results were obtained (Figure 4.20):



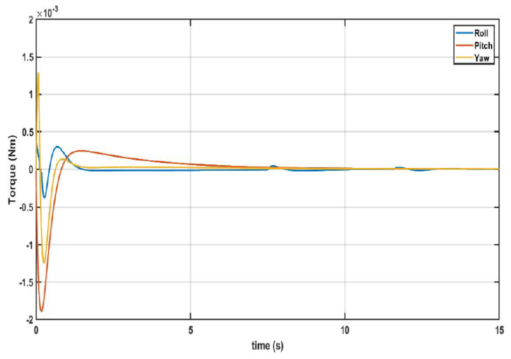
(a) Time response in front of a unitary step. **(b)** Time response in front of a unitary slope ramp.

Figure 4.20 Control algorithm responses obtained by using Root locus method. Source: [33]

Regarding these graphs, an undesired stationary error was detected (Figure 4.20b), which will require the later implementation of a static error correction. By applying an adaptative control in order to adjust the KPp to the system behavior depending on the error signal |u|, the results presented below are obtained (Figure 4.21):



(a) Alignment error in Euler Angles.



(b) Calculated torque signal by controller.

Figure 4.21 Adaptive control obtained results graphs. Source: [33]

According to the already introduced Joan Serrano's cascade control loop (Figure 4.18), and after observing the obtained results (Figures 4.20,4.21), it can be concluded that this control algorithm not only can be applied to the prototype design that is being developed in this project but also is the optimum solution for the present prototype measuring capabilities and code characteristics.

4.3 Printed Circuit Board (PCB)

Once all the proposed improvements have been studied, justified, and properly tested, the final solution for the ADCS subsystem can be implemented in the final circuit. So as to offer a clear explanation for these modifications over the already designed ADCS PCB developed for previous students (such as Jordan Morales or Miguel Roguer)[34], this chapter of the report will be divided into the following fragments:

4.3.1 Stepper motor integration and interconnection

Regarding the already studied deployable mechanism applied to the origami-based solar panel, the power supply and control of the stepper motor that will trigger the deploying movement is of paramount importance for the performance of the whole prototype.

Having said that, and taking into account the nature of the electronic components used for the implementation of this system, the following electronic diagram will be required (Figure 4.22):

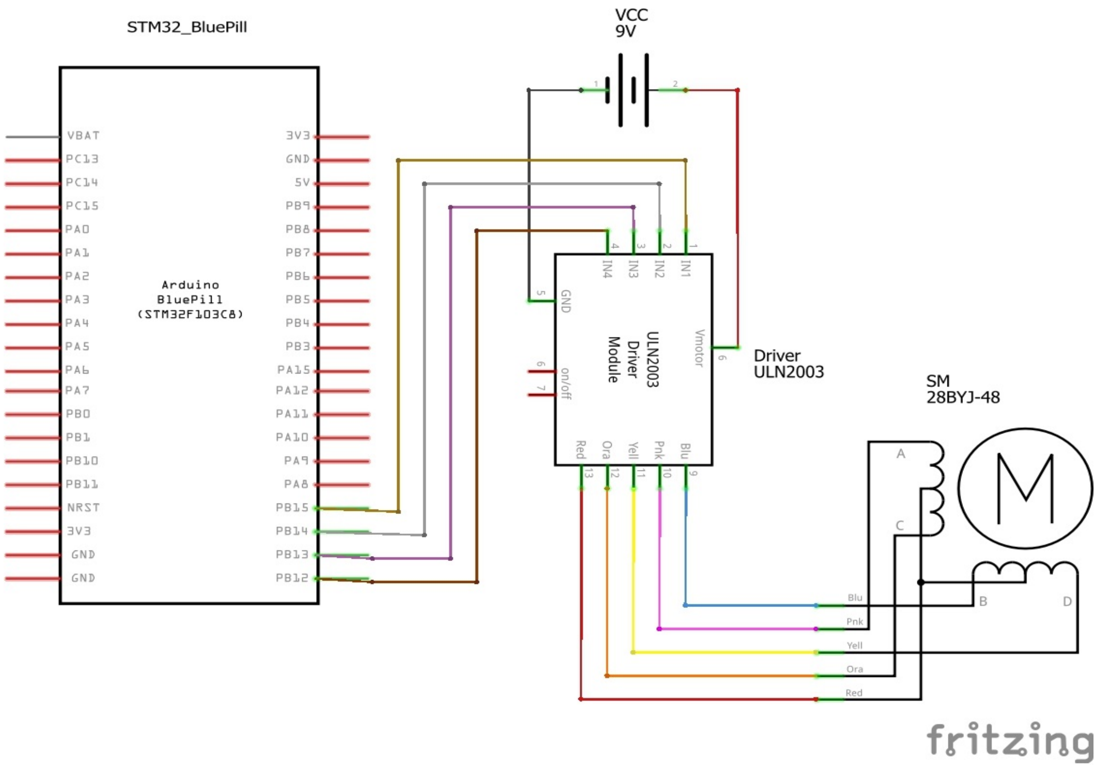


Figure 4.22 Stepper motor's connection electric diagram. Source: Own

After seeing the previous image, the following color code table must also be considered (Table 4.9):

ULN2003 Driver	28BYJ-48 Stepper Motor	Bluepill pin	Color Code
IN1	-	B15 pinout	Ocher
IN2	-	B14 pinout	Grey
IN3	-	B13 pinout	Purple
IN4	-	B12 pinout	Brown
Blu	Blu	-	Blue
Pnk	Pnk	-	Pink
Yell	Yell	-	Yellow
Ora	Ora	-	Orange
Red	Red	-	Red
V motor	Connected to 9V battery		Red
GND	Connected to 9V battery		Black

Table 4.9 Stepper motor's connection color code. Source: Own

4.3.2 Implemented modifications

Concerning the proposed improvements over the measurement process, seen in the previous chapters, and the already seen implementation of the torque generation system, it's now time to document all the modifications that will be implemented over the final PCB prototype.

4.3.2.1 List of modifications

- **MPU DMP access bus modification:**
  As already seen and explained, the access to the DPM (Digital Motion Processor) of the MPU requires a change in the used communication bus from the SPI communication bus to the I^2C bus. This change will be seen in the final PCB scheme (Figure 4.23).
- **MPU calibration connection modification:**
  The new calibration method proposed for this new prototype requires the connection of MPU's pins according to the already represented scheme (see 4.8b scheme).
- **Stepper motor integration:**
  The addition of the solar panel deployable subsystem to the final prototype requires the microcontroller to be connected to the stepper motor in order to send the required pulse sequence to the motor's inductances. The performance of this connection will be seen in the final PCBs connections scheme (Figure 4.23).
- **Motor driver VIN and GND rectification:**
  In the original design, it was found that the VIN and GND signals from the motor driver connections were switched, this fact lead to the consequence of the addition of some additional wires in order to solve this problem. In this proposed redesign, this mistake will be solved.

Once the list of modifications has been briefly explained and implemented over the final design, the following connection scheme of the final PCB design is obtained (Figure 4.23):

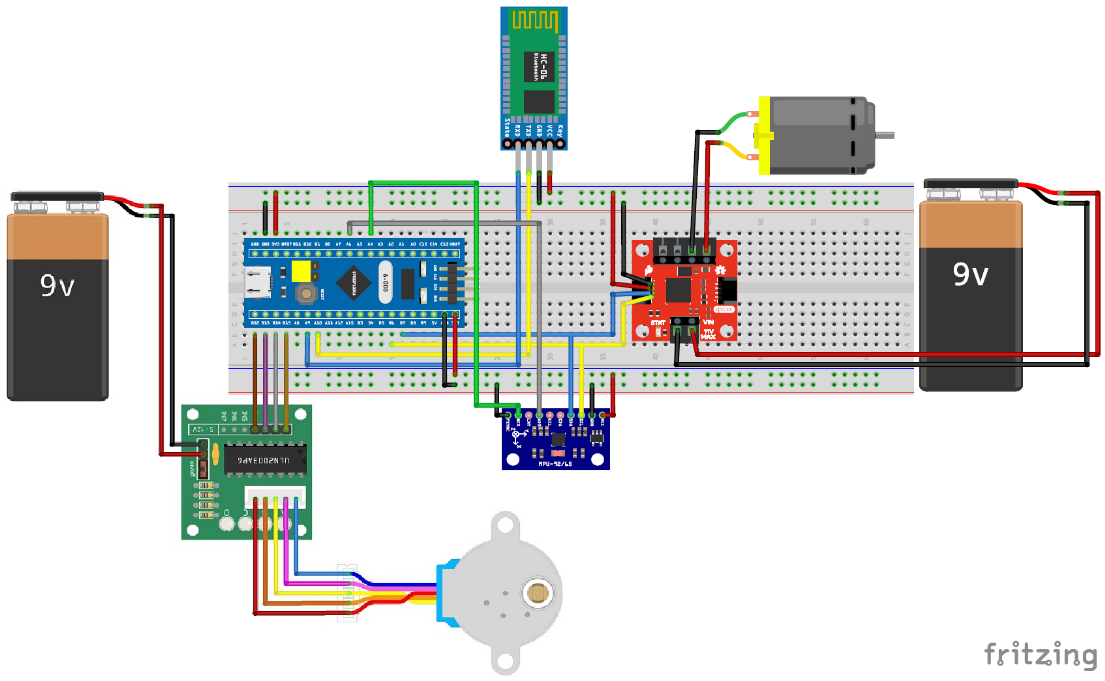


Figure 4.23 Figurative representation of connections of the whole system. Source: Own

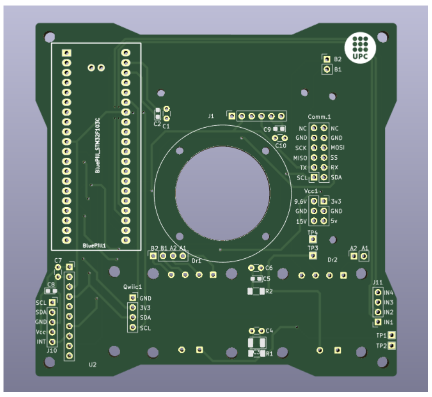
The components used in the previous image are enumerated in the following table 4.10:

Component	Commercial model
9V Battery	ENEGON SM001 Battery
Microcontroller	STM32 Bluepill
Bluetooth module	HC-06 BT module
Inertial measurement unit (IMU)	MPU-9250
DC motor driver	SparkFun Qwiik Rob-15451
Stepper motor driver	ULN2003 Driver board
DC motor	RF-300EA
Stepper motor	28BYJ-48 Stepper motor

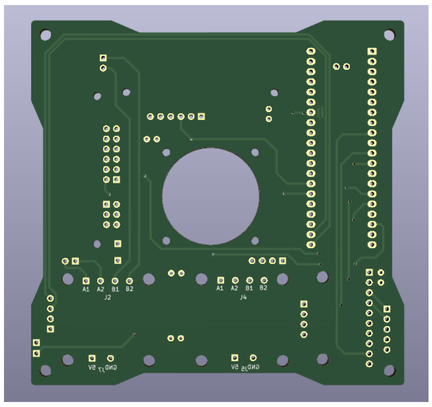
Table 4.10 Used hardware components list. Source: Own

4.3.3 Proposed PCB redesign

After the implementation of all the already mentioned modifications, and in order to exemplify the redesigned PCB tracks, the Printed Circuit Board presented below is obtained (Figure 4.24):



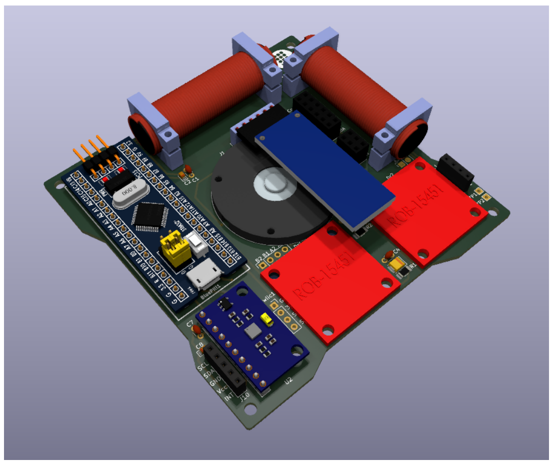
(a) Redesigned PCB top view (without components).



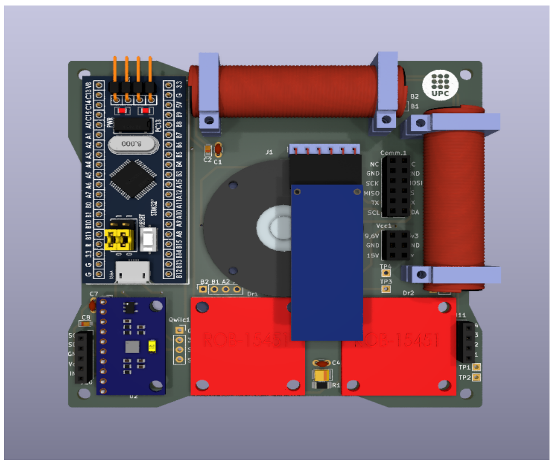
(b) Redesigned PCB underside (without components).

Figure 4.24 Redesigned PCB 3D model (without components). Source: Own

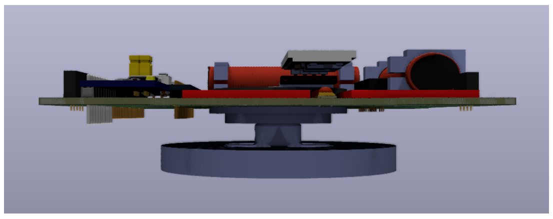
After fixing the required components over this redesign (always respecting the higher possible degree of compactness) the final PCB redesign becomes as follows (Figure 4.25):



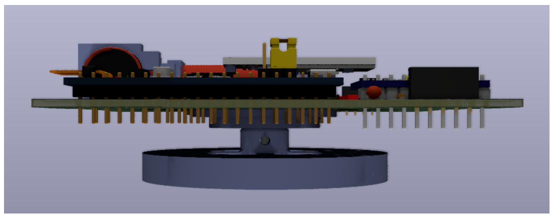
(a) Redesigned PCB isometric view (with components).



(b) Redesigned PCB top view (with components).



(c) Redesigned PCB front side view (with components).



(d) Redesigned PCB left side view (with components).

Figure 4.25 Redesigned PCB 3D model (with components). Source: Own

4.3.3.1 Comparative table

To conclude this chapter and aiming to perform a good summary of the improvements applied to both the programming code and the final PCB redesign, the following table 4.11 has been performed:

	Original device	**Final prototype design**
MPU measurement method	Complementary filter applied to sensor measurements	Digital motion processor (DMP)
Communication bus for MPU measurement	Serial peripheral interface (SPI)	Inter-Integrated Circuit (I^2C)
MPU calibration method	Complex multiple axis rotation calibration	Simple automatic calibration by code execution
Rectification of the VIN and GND PCB tracks	Not implemented	Implemented
Coarse positioning used impulse	Complex uncontrolled impulse setting	Simple step controlled impulse setting
Studied ADCS control algorithm	Single PID controller tuning	Cascade multiple controller control loop
Stepper motor hardware integration	Not implemented	Implemented

Table 4.11 Applied improvements comparative table. Source: Own

Chapter 5

Integration of all subsystem in the final prototype

Once both deployable solar panel subsystem and ADCS printed circuit board has been studied, developed, and manufactured, it is now the time to study and justify the optimum way of integrating all these systems in one single prototype. In this chapter, the final demonstration of the proper behavior of the whole prototype will be also included by means of the implementation and graphical exemplification of some lately described tests. Aiming to do a proper explanation of this implementation, the next fragment of this report will be divided into the following parts:

5.1 Solar panel subsystem

The objective of this section will be the explanation and justification of the integration of the already developed and explained deployable solar panel subsystem. This fragment will include the mechanical-oriented integration (including the parts designed in order to do this) and the electronic part (including the integration of the stepper motor electronics).

5.1.1 Deployment system

The already designed solar panel deployable system will be fixed to the CubeSat frame by using the screws shown in the following image (Figure 5.1):

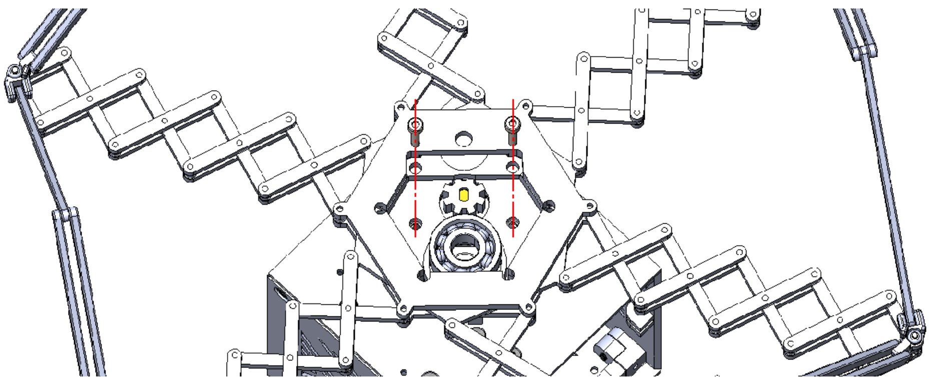
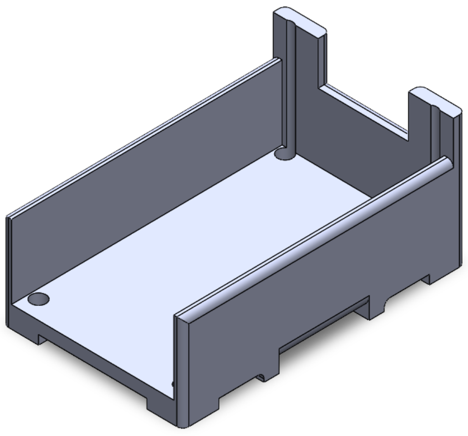


Figure 5.1 Deployable mechanism screw fixation. Source: Own

It must also be mentioned that the same screws will cross the CubeSat wall and also fix the stepper motor to the internal side of the wall.

5.1.2 Power supply and control system

Aiming to a proper fixation of the electronic components related to the stepper motor circuit (already detailed in previous chapters), the following pieces have been designed (Figures 5.2,5.3):


(a) Battery support isometric view.

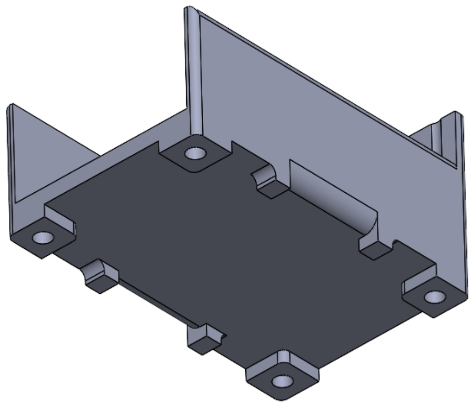

(b) Battery support bottom view.

Figure 5.2 Battery support 3D model. Source: Own

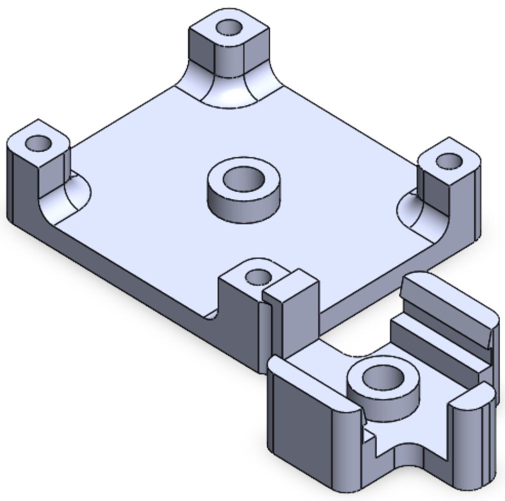


Figure 5.3 Driver + DC converter support 3D model. Source: Own

By using the previously represented parts (Figures 5.2,5.3), the assembling of the different components related to the power supply and the control of the stepper motor has been implemented as follows (Figures 5.4,5.5):

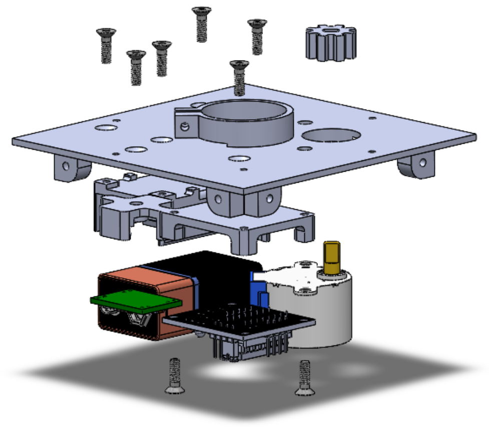
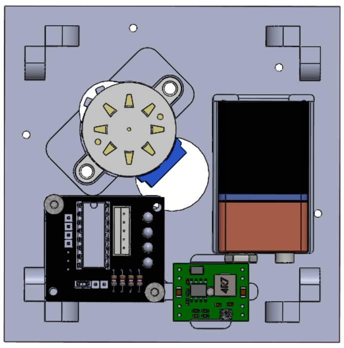
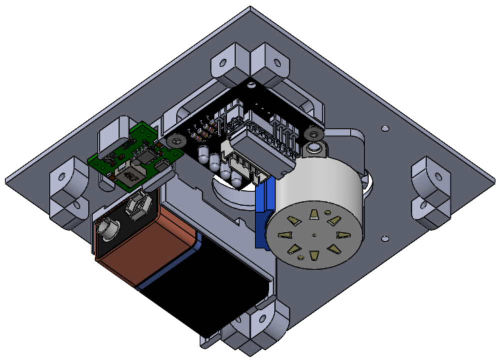


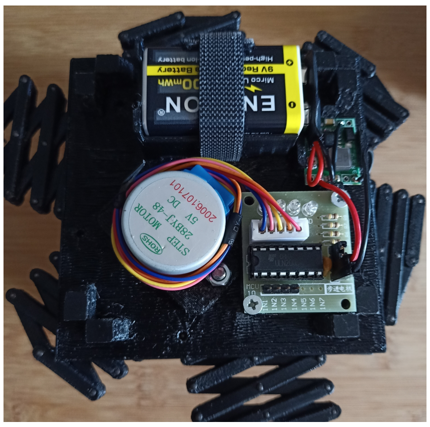
Figure 5.4 Deployable mechanism control wall exploded view. Source: Own



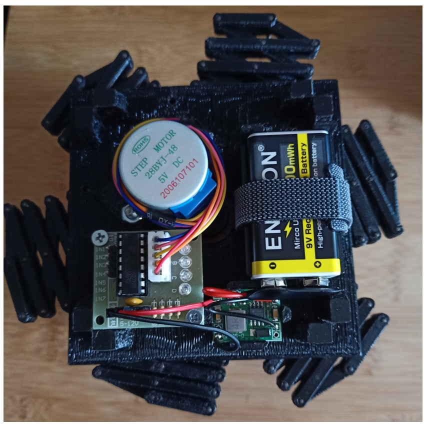
(a) Deployable mechanism control wall bottom view.



(b) Deployable mechanism control wall isometric view.



(c) Real deployable mechanism control wall underside.



(d) Real deployable mechanism control wall underside.

Figure 5.5 Deployable mechanism control wall. Source: Own

5.2 PCB subsystem

In this fragment of the chapter, the integration of the ADCS PCB device will be mechanically justified. Moreover, the modification of some of the components of the PCB and the relation between this ADCS device and the RW specifications will be also presented.

5.2.1 ADCS and communication circuit

After the development and assembly of the ADCS PCB components, the integration of this subsystem into the final CubeSat prototype will be displayed by the redesign of some of the CubeSat walls. The slight modification performed to the wall in order to properly integrate this new component is represented in the following image 5.6:

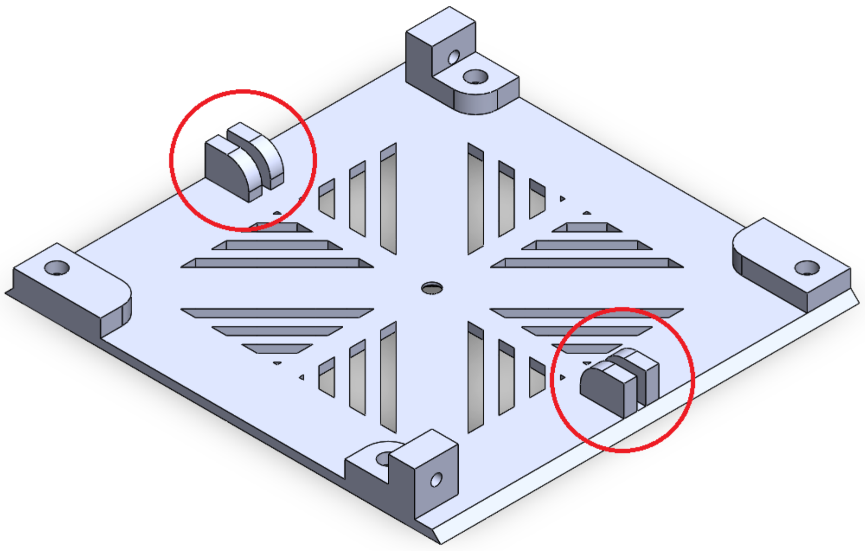


Figure 5.6 Side wall's PCB integration modification 3D model. Source: Own

By using these additional structures, the ADCS PCB can be easily integrated (Figure 5.7).

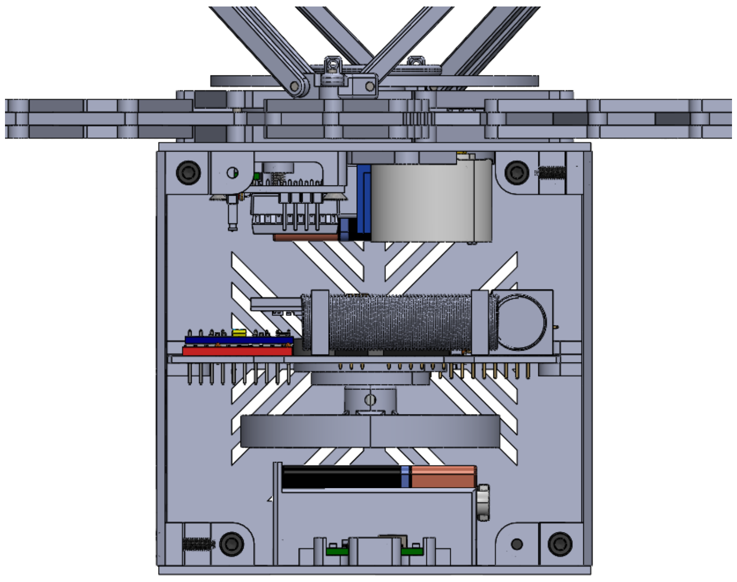
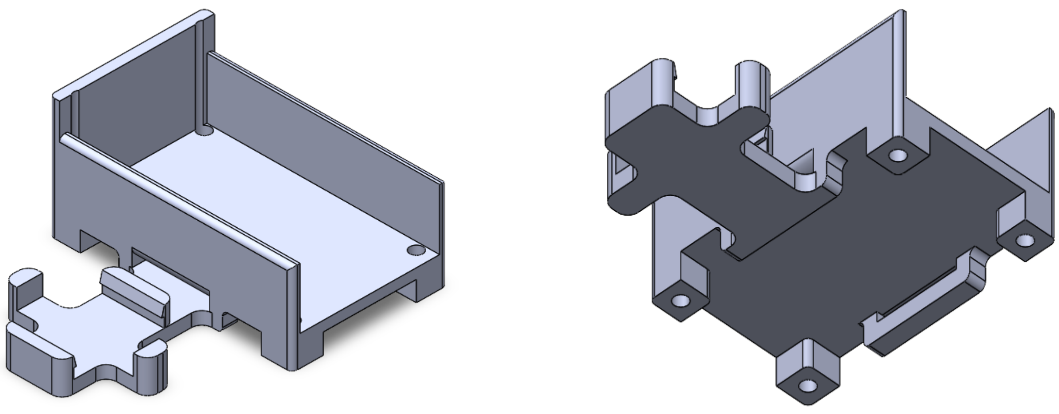


Figure 5.7 Final CubeSat prototype 3D model internal view. Source: Own

5.2.2 Power supply system

Regarding the upgrade of the power supply system of the ADCS PCB, to enhance the compactness of the whole prototype, the battery pack has been integrated into the prototype by using the following piece (Figure 5.8):



(a) Battery + DC converter support isometric view. **(b)** Battery + DC converter support underside view.

Figure 5.8 Battery + DC converter support 3D model. Source: Own

By fixing this piece to the bottom wall of the final prototype, the power supply circuit can be integrated into the assembly (Figure 5.9).

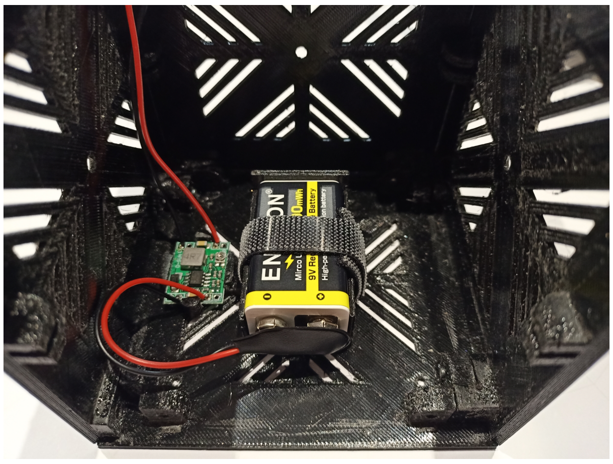


Figure 5.9 Real power supply system fixation on CubeSat prototype. Source: Own

5.3 Reaction Wheel (RW)

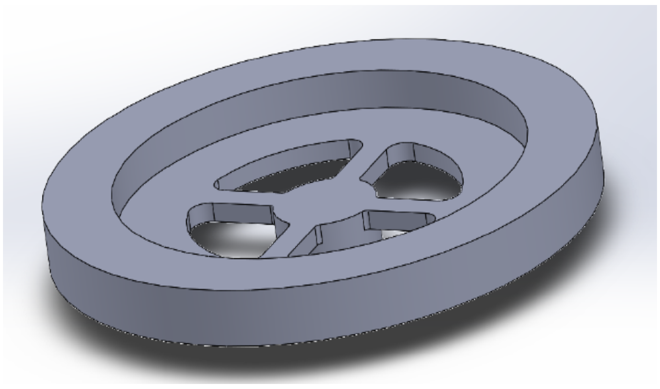
The main objective of this fragment is to explain the design restriction that all the other subsystems impose on the design of the reaction wheel (in terms of both dimensions and shape). This paragraph will also explain the integration of the prototype into the air-bearing test device.

To begin with, it must be mentioned that, because of the integration of all the CubeSat subsystems in one prototype, the optimization of the inner volume of the device is of paramount importance. Having said that, the final dimensional characteristics of the reaction wheel (RW) will be much smaller than they were in previous prototypes, as it can be observed in the following comparison table (Table 5.1):

Part	**Parametr**	**Original RW value (mm)**	**New RW value (mm)**
Disc radius	r_{disc}	60	20
Ring radius	r_{ring}	80	30
Hollow radius	r_{hollow}	12,5	7,5
Cylinder radius	r_{cyl}	-	6
Disc height	h_{disc}	2,5	2,5
Ring height	h_{ring}	7,5	7,5
Hollow height	h_{hollow}	2,5	2,5
Cylinder height	h_{cyl}	-	7

Table 5.1 Original/Modified reaction wheels' dimensions. Source: Own

Regarding Table 5.1 and according to the hardware-focused solution presented in Section 4.2.2.1, a comparative calculation of the momentum of inertia of the Reaction Wheels is required both to justify the presented solution and to exemplify the compactness restriction of the final prototype.


(a) Reaction Wheel's original design.

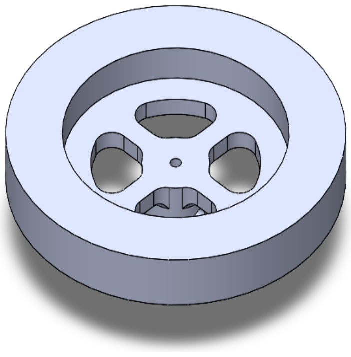

(b) Reaction Wheel's improved design.

Figure 5.10 Original/Modified Reaction Wheel's design. Source: Own

Considering the geometrical characteristics of the designed reaction wheel (Figure 5.10) and the following value of PET density, the RW's mass calculation can be performed as follows:

$$\rho_{PET} = 1370 kg/m^3 \tag{5.1}$$

$$\begin{aligned}\mathbf{m_{hollow}} &= \rho_{PET} \cdot \pi r_{hollow}^2 \cdot h_{hollow} = \\ &= 1370 kg/m^3 \cdot \pi \cdot 0,0075^2 m \cdot 0,0025 m = \mathbf{0,605 \cdot 10^{-3} kg}\end{aligned} \tag{5.2}$$

$$\begin{aligned}\mathbf{m_{disc}} &= \rho_{PET} \cdot \pi r_{disc}^2 \cdot h_{disk} - m_{hollow} = \\ &= 1370 kg/m^3 \cdot \pi \cdot 0,020^2 m \cdot 0,0025 m - 0,605 \cdot 10^{-3} kg = \mathbf{3,699 \cdot 10^{-3} kg}\end{aligned} \tag{5.3}$$

$$\begin{aligned}\mathbf{m_{cyl}} &= \rho_{PET} \cdot \pi r_{cyl}^2 \cdot (h_{cyl} + h_{disc}) = \\ &= 1370 kg/m^3 \cdot \pi \cdot 0,006^2 m \cdot (0,007 m + 0,0025 m) = \mathbf{1,472 \cdot 10^{-3} kg}\end{aligned} \tag{5.4}$$

$$\begin{aligned}\mathbf{m_{ring}} &= \rho_{PET} \cdot \pi (r_{ring}^2 - r_{disc}^2) \cdot h_{ring} = \\ &= 1370 kg/m^3 \cdot \pi \cdot (0,030^2 m - 0,020^2 m) \cdot 0,0075 m = \mathbf{16,140 \cdot 10^{-3} kg}\end{aligned} \tag{5.5}$$

$$\begin{aligned}\mathbf{m_{RW}} &= m_{disc} + m_{cyl} + m_{ring} = \\ &= 3,699 \cdot 10^{-3} kg + 1,472 \cdot 10^{-3} kg + 16,140 \cdot 10^{-3} kg = \mathbf{21,311 \cdot 10^{-3} kg}\end{aligned} \tag{5.6}$$

Then, the momentum of inertia can be determined as:

$$\mathbf{I_{disc}} = \frac{m_{disc} \cdot r_{disc}^2}{2} = \frac{3,699 \cdot 10^{-3} kg \cdot 0,020^2 m}{2} = \mathbf{7,398 \cdot 10^{-7} kg \cdot m^2} \tag{5.7}$$

$$\mathbf{I_{cyl}} = \frac{m_{cyl} \cdot r_{cyl}^2}{2} = \frac{1,472 \cdot 10^{-3} kg \cdot 0,006^2 m}{2} = \mathbf{2,649 \cdot 10^{-8} kg \cdot m^2} \tag{5.8}$$

$$\mathbf{I_{ring}} = \frac{m_{ring} \cdot (r_{ring}^2 + r_{disc}^2)}{2} = \frac{16,140 \cdot 10^{-3} kg (0,030^2 m + 0,020^2 m)}{2} = \mathbf{1,049 \cdot 10^{-5} kg \cdot m^2} \tag{5.9}$$

$$\begin{aligned}\mathbf{I_{RW}} &= I_{disc} + I_{cyl} + I_{ring} = \\ &= 7,398 \cdot 10^{-7} kg \cdot m^2 + 2,649 \cdot 10^{-8} kg \cdot m^2 + 1,049 \cdot 10^{-5} kg \cdot m^2 = \\ &= \mathbf{1,126 \cdot 10^{-5} kg \cdot m^2}\end{aligned} \tag{5.10}$$

Once these inertial momentum values have been obtained and comparing theme with the ones extracted from Yi's thesis [1], the following reduction can be observed:

$$I_{RW,origin} = 4.693 \cdot 10^{-4} kg \cdot m^2 \tag{5.11}$$

$$\begin{gathered}\textbf{Inertia Percentage Change} = \tfrac{1{,}126 \cdot 10^{-5}}{4.693 \cdot 10^{-4}} \cdot 100 = 2{,}399 \\ \textbf{Reduction Ratio} = 100 - 2{,}399 = \mathbf{97{,}601\%}\end{gathered} \tag{5.12}$$

According to the previous result (Value 5.12), it can be concluded that the introduced hardware-focused solution not only will fit the exposed control advantage but also will fit the dimensional requirements of the final prototype.

5.4 Software integration

This section of the project will be mainly dedicated to the explanation and justification of the integration of the developed devices focusing on its software and code-related aspects. In this section, some consideration of the mechanical implemented devices will be also taken into account.

5.4.1 Stepper motor pulse determination

In order to ensure the proper, complete, and smooth deployment of the developed and manufactured Origami-Based solar panel, and according to the stepper motor operating principle (See Chapter 3), it is of paramount importance to calculate the required pulse number, which must be defined in the deployment code.

Regarding some geometrical considerations of the deployment mechanism, the following list of variables can be obtained (Table 5.2):

Description	Parameter
Pinion-gear number of teeth	n_{Pgear}
Crown-gear number of teeth	n_{Cgear}
Reduction ratio	red_{ratio}
Stride angle of the stepper motor	α_{step}
Desired crown-gear angle rotation	$\alpha_{rotation}$
Pinion-gear angle rotation	α_{Pgear}
Number of pulses	n_{pulses}

Table 5.2 Step calculation parameter table. Source: Own

According to the previously presented technical performance of the stepper motor 28BYJ-48 (Table 5.2) and the fact that the Pulse/Step ratio is 8, the required pulse number is calculated as follows:

$$\begin{aligned} &\alpha_{rotation} = 40^o \\ &n_{Pgear} = 8 \\ &n_{Cgear} = 36 \\ &red_{ratio} = 4,5/1 \\ &\alpha_{step} = 5,625 \cdot \tfrac{1}{64}{}^o/step \end{aligned} \tag{5.13}$$

Once the desired rotation of the crown-gear is known, the pinion-gear required rotation can be determined:

$$\alpha_{Pgear} = \alpha_{rotation} \cdot \frac{n_{Cgear}}{n_{Pgear}} = 40^o \cdot \frac{36}{8} = 180^o \tag{5.14}$$

$$n_{pulses} = \frac{\alpha_{Pgear}}{8 \cdot \alpha_{step}} = \frac{180^o}{8 \cdot 5,625 \cdot \frac{1}{64}{}^o/step} = \mathbf{256pulse} \tag{5.15}$$

Therefore, the number of pulses required for the complete deployment of the Origami-Based solar panel mechanism is hard-coded in the final code. Despite this fact, it must be mentioned that the original consideration of 40^o has been determined through the observation of the mechanism.

5.4.2 Prototype code integration

To further extend the available functionalities of the hardware and their integration and control through software inputs, numerous modifications of the code have been implemented in the prototype control code. Therefore, the new operation flowchart of the function mode selection can be observed at the following figure (Figure 5.11):

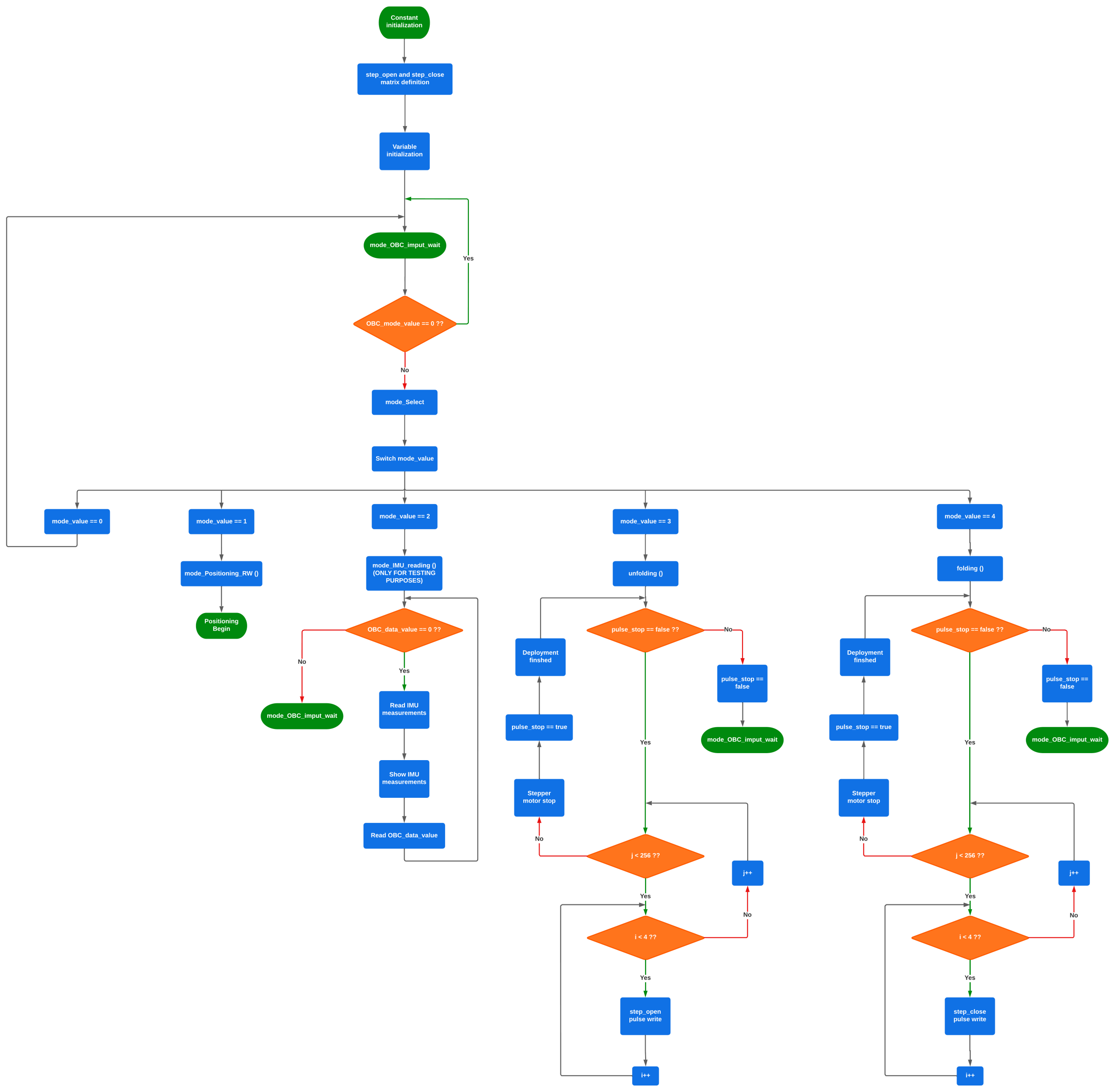


Figure 5.11 Mode Selection code flowchart (Augmented version in Appendix D.1). Source: Own

5.5 Final tests

Finally, in order to experimentally illustrate the achieved performance of all the developed subsystems, the different operation modes developed along this project have been tested. Consequently, each one of these tests is focused to evaluate the behavior of one independent functionality of the studied subsystems, such as the deployment of the solar panel mechanism, the calibration of the MPU, or the overall accuracy of the coarse pointing mode.

5.5.1 Origami-based solar panel deployment

The first performed test was a deployment of the complete solar panel mechanism in order to both determine the accuracy of its performance and ensure the integrity of its internal components after the final assembly.

After the execution of this test, the following images were taken to illustrate its correct behavior (Figure 5.12).

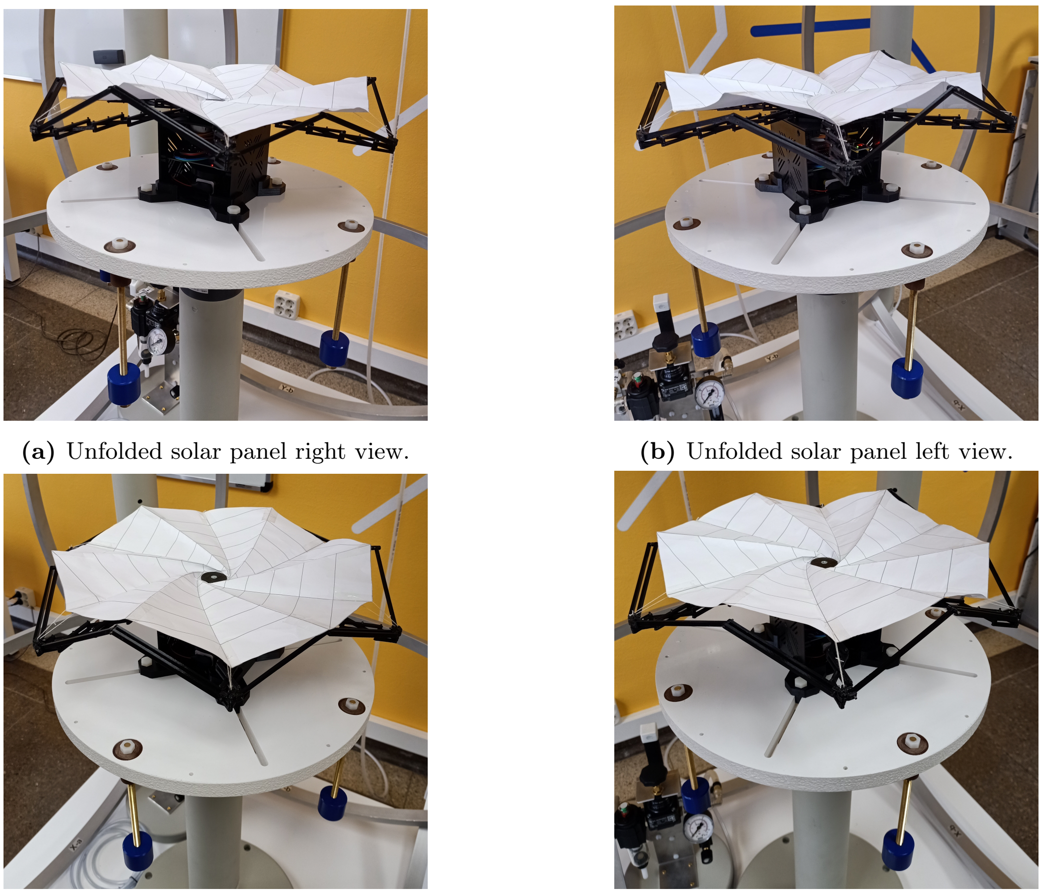

(a) Unfolded solar panel right view.
(b) Unfolded solar panel left view.
(c) Unfolded solar panel top right view.
(d) Unfolded solar panel top left view.

Figure 5.12 Deployable mechanism control wall. Source: Own

In the following link, a video of the test process is displayed in order to ease a proper comprehension of the subsystem's behavior.

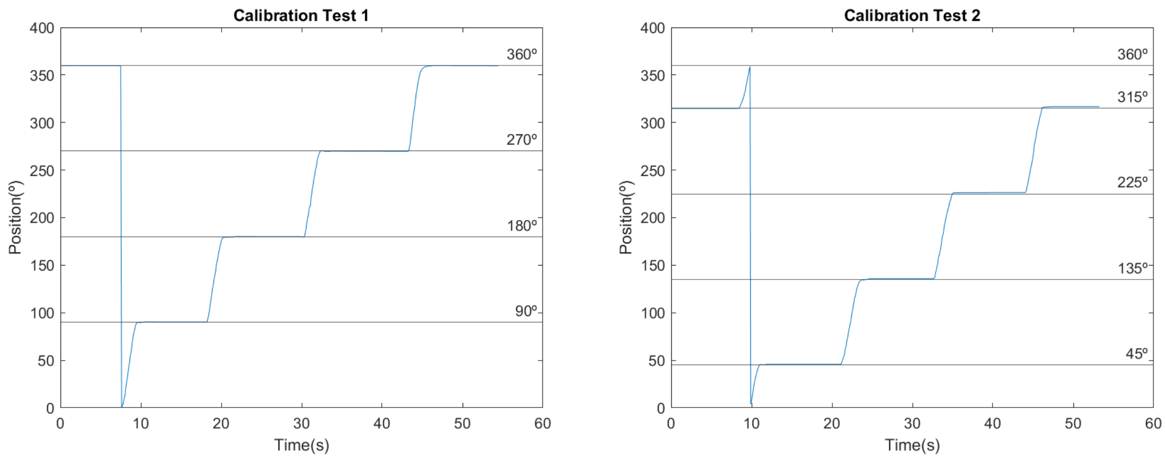
https://drive.google.com/file/d/1_y8mEyIsTV5MP7YXK1vAHDN82ubxzElN/view?usp=sharing

5.5.2 MPU calibration test

Once proved the proper behavior of the Origami-based solar panel deployment subsystem, and prior to the final ADCS coarse control tests, the accuracy achieved by using the DMP access and the new calibration method must be also proved.

In order to test its behavior, 90º movements of the final calibrated prototype were performed and registered to ensure the proper orientation recognition of the improved subsystem, since this type of movements are simple enough to be tracked with external measuring tools, and the comparison between the registered data and the expected result allow to extract a trustful enough figure of merit of the calibration accuracy.

The results corresponding to the testing movements performed can be seen in Figure 5.13:



(a) Calibration test results (No pass trough zero). **(b)** Calibration test results (Pass trough zero).

Figure 5.13 Calibration test results. Source: Own

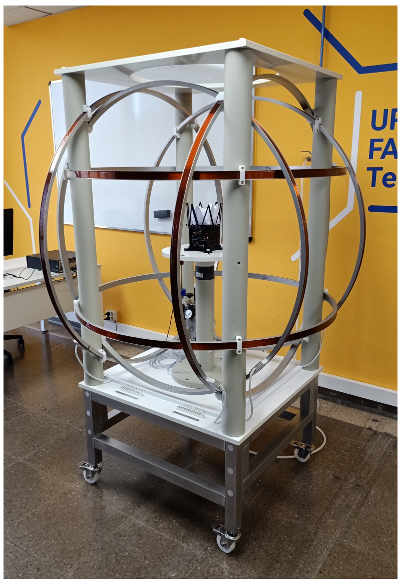
Considering figure 5.13, and according to the 90º performed movements, it can be concluded the final measurement performance of the prototype is accurate enough to precisely track the movements experienced by the system. Therefore demonstrating a reliability of the movement tracking not observed at previous iterations of the ADCS subsystem.

In the previous figure 5.13b, it can also be observed the effect of passing through the 0º position while the device is operating, since the rectification of this sudden jump had to be implemented in the final code so as to clarify the data reading.

5.5.3 ADCS coarse control test

5.5.3.1 Test Setup

Figures 5.14 shows the professional Air Bearing used in the project. This testbed includes a pivoting platform that allows the free movement of the hosted prototypes, consequently enabling the simulation of nanosatellite movement self-control and the design of CubeSat 1U units fully equipped to control their own spatial orientation (see Appendix F.6).


(a) Air bearing system right view.

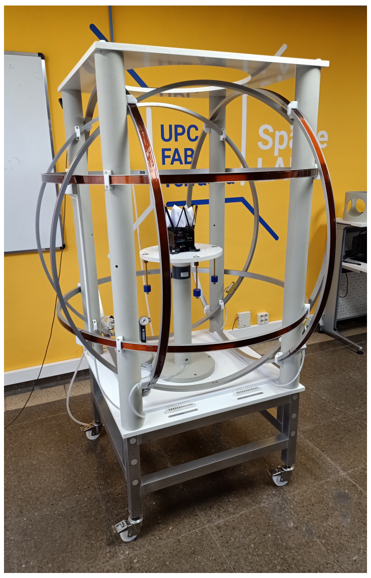
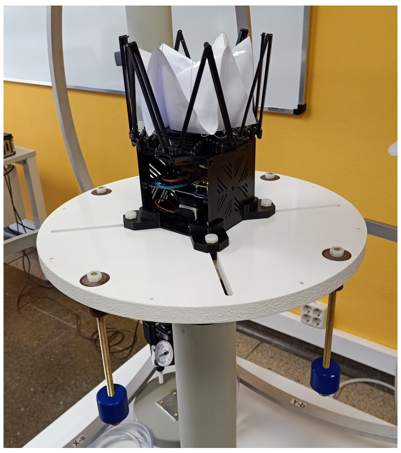

(b) Air bearing system left view.

Figure 5.14 Air bearing real system. Source: Own

An air-flow conditioning unit is located below the air bearing, and it is connected to a compressed air line which blows air into the structure, consequently creating a near-zero friction environment system once the CubeSat is placed on the Air bearing platform. This system will then allow to reproduce the CubeSat conditions in space by isolation from the surrounding of the system represented at Figure 5.15.


(a) CubeSat attached to air bearing system right view.

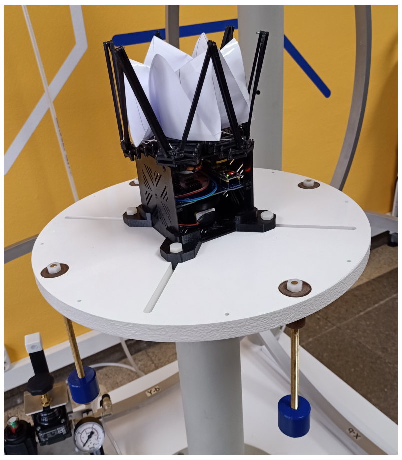

(b) CubeSat attached to air bearing system left view.

Figure 5.15 CubeSat attached to real air bearing system . Source: Own

Considering so, it is required to perform a detailed calculation of the momentum of inertia of the whole air pivoting body (including both platform and CubeSat prototype) before performing any test with this equipment.

So as to do this calculation, some approximations over the air pivoting platform will be taken into account.

- The four counterpoises and their threaded rods will be taken as uniformly distributed weight rings placed under the pivoting platform.
- The whole CubeSat prototype will be taken as a 1,5U uniformly weight distributed CubeSat.

According to these considerations, the following data can be used:

Part	Parameter	Value
Platform radius	r_{plat}	200mm
Counterpoises internal radius	r_{icount}	170mm
Counterpoises external radius	r_{ecount}	190mm
Platform height	h_{plat}	19mm
Counterpoises weight	m_{count}	330·4 g
CubeSat weight	m_{Cube}	582g
PVC density	ρ_{PVC}	$200kg/m^3$

Table 5.3 Air bearing inertia calculation data table. Source: Own

Considering the previous data, the inertia calculations can be finally performed as follows:

$$\begin{aligned}\mathbf{m_{plat}} &= \rho_{PVC} \cdot \pi r_{plat}^2 \cdot h_{plat} = \\ &= 200kg/m^3 \cdot \pi \cdot 0,200^2 m \cdot 0,019m = \mathbf{0,477kg}\end{aligned} \tag{5.16}$$

$$\mathbf{I_{plat}} = \frac{m_{plat} \cdot r_{plat}^2}{2} = \frac{0,477kg \cdot 0,200^2 m}{2} = \mathbf{9,54 \cdot 10^{-3} kg \cdot m^2} \tag{5.17}$$

$$\mathbf{I_{cont}} = \frac{m_{cont} \cdot (r_{icont}^2 + r_{econt}^2)}{2} = \frac{1,2kg \cdot (0,170^2 m + 0,190^2 m)}{2} = \mathbf{0,039kg \cdot m^2} \tag{5.18}$$

$$\mathbf{I_{Cube}} = \frac{m_{Cube} \cdot (s_{Cube}^2 + h_{Cube}^2)}{12} = \frac{0,582kg \cdot (0,10^2 + 0,15^2)}{12} = \mathbf{1,576 \cdot 10^{-3} kg \cdot m^2} \tag{5.19}$$

$$\begin{aligned}\mathbf{I_{total}} &= I_{plat} + I_{cont} + I_{Cube} = \\ &= 9,54 \cdot 10^{-3} kg \cdot m^2 + 0,039 kg \cdot m^2 + 1,576 \cdot 10^{-3} kg \cdot m^2 = \\ &= \mathbf{50,116 \cdot 10^{-3} kg \cdot m^2}\end{aligned} \tag{5.20}$$

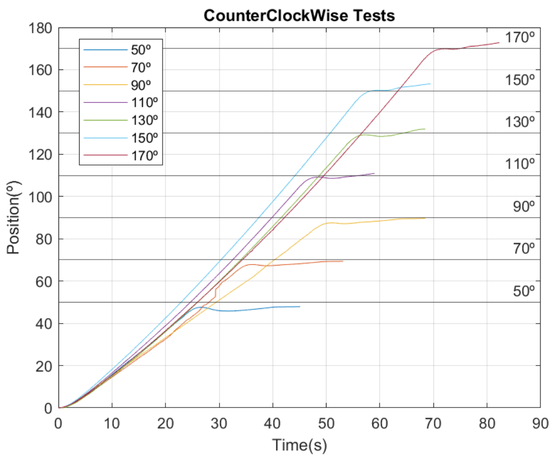
In order to do a proper interpretation of the obtained momentum of inertia value, it must be taken into account that an excessively high value of this parameter could lead to the malfunction of the whole pointing mode as a consequence of the difficult modification of the movement speed of the measurement body, both at acceleration and deceleration. Additionally, it must be also considered that the previous process could become uncontrolled if the momentum of inertia value is too low, since it could reach excessively high speeds once the impulse on the RW is triggered.

5.5.3.2 Test Results

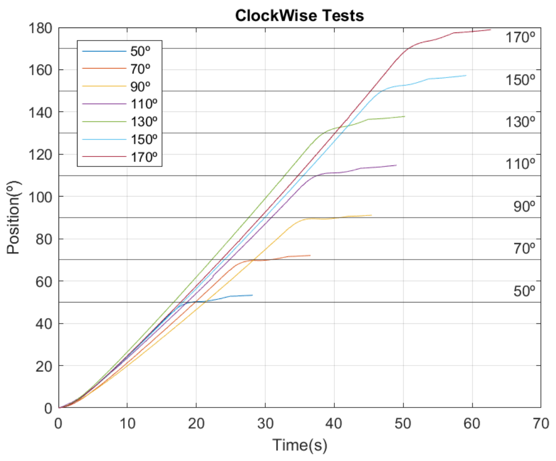
Once the setup used for these tests has been properly presented, and some mechanical characteristics of the prototype and air-bearing testbed have been justified, the final coarse mode operating tests are performed. Several maneuvers have been performed to ensure the correct behavior of the spacecraft's ADCS subsystem in the available movement processes that the device can perform (clockwise, counterclockwise, large rotations, etc.).

5.5.3.2.1 Clockwise and counterclockwise small rotations:

Considering that the most common rotations that this device will experience will be smaller than 180º rotations, the response of the ADCS subsystems to these settings must be studied both for clockwise rotations and counterclockwise rotations, which leads to the following measurement results:



(a) Counterclockwise rotation test results.



(b) Clockwise rotation test results.

Figure 5.16 Small rotation test results. Source: Own

According to the previously seen Figure 5.16, and comparing clockwise rotations (represented in Figure 5.16b) and counterclockwise rotations (represented in Figure 5.16a), it can be concluded that the maneuvers in both rotation directions behave correctly. However, it is observed that the deceleration stage of the clockwise maneuvers is not able to properly stop the rotation of the device. This fact, despite not being of great importance given that the residual movement could be corrected by action of the fine-pointing mode, can be explained by parasite torque generated for a turbulent air flux in the air-bearing device.

5.5.3.2.2 Clockwise and counterclockwise large rotations:

When talking about the behavior of the device when large rotations (bigger than 180º) maneuvers are performed, and according to the programmed software code, it must be mentioned that the device will always perform the shortest possible rotation movement.

In order to graphically represent this setting, some tests were performed, obtaining the following results (Figure 5.17):

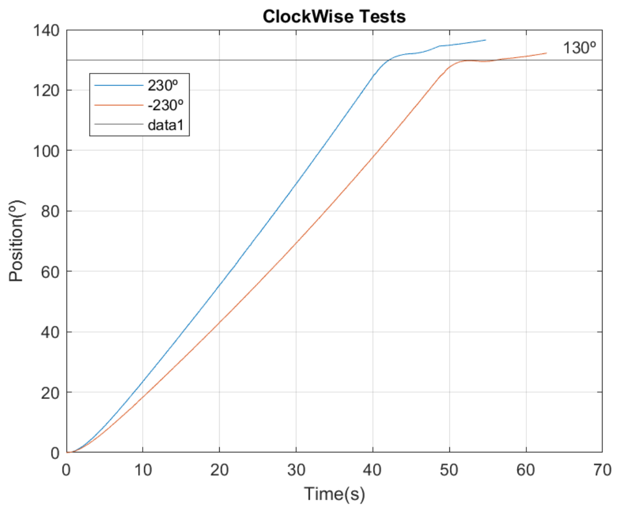


Figure 5.17 Large rotation test results. Source: Own

As it can be seen in figure 5.17, the displayed movement after the input of a maneuver command of 230º in either rotation direction is the rotation of 130º in opposite direction, being this the shortest possible rotation movement to place the device in the target position. Consequently, it can be affirmed that the final behavior of the device for large movements has achieved the desired requirements.

It must also be mentioned that the previously commented parasite torque for counterclockwise rotations can be also seen in these maneuvers.

Chapter 6

Non-technical documents

The main objective of this chapter is to compile all the documents not strictly related to the development and manufacturing of the CubeSat technology studied during this project. The documents included in this chapter will be dedicated to the representation of temporal planning, an economic study of the project, and a study of the environmental impact that the used tools and applied processes exert.

6.1 Planning of the project

This section illustrates, enumerates, and describes all bibliographic research phases, processes used for the development and manufacturing of the CubeSat prototype, as well as the explanation and justification stages of the final report. In this section, a Gantt diagram will also be included to graphically exemplify the programmed schedule and deadlines in the elaboration of the project.

6.1.1 Task identification

In the table 6.1 each one of the performed tasks is enumerated, described, and represented with its respective duration time estimation and order dependency.

1. **Introduction:** A first approach to the CubeSat spacecraft and its subsystems, this part is meant to emphasize the solar panels and the attitude control subsystems.
2. **Searching and analysis of bibliography:** Information research in order to develop a more solid concept of the studied systems.
3. **Initial concept to be developed about solar arrays based on origami patterns:** Starting from the already searched information, develop a more defined work plan about solar panels subsystems.
4. **Development of the initial concept about attitude control applied to CubeSat technologies** Starting from the already searched information, develop a more defined work plan about attitude control subsystems.
5. **Analysis of the solar array unfolding methods:** Study different possible methods of deployment to the solar panel system. Brainstorm different mechanisms to obtain the optimum result.
6. **Development of a solid concept about the proper solution applied to solar arrays:** Once the information research has been done, the next step is to define a solid concept to give a solution to the proposed problem. This concept will be defined as a first 3D model of a possible solution.

7. **Design a preliminary prototype to determine the technical viability of the idea:** Manufacturing the 3D models defined in the previous step could be necessary in some cases to ensure the correct functionality of the proposed preliminary solution.
8. **Testing of the proposed solution applied to solar arrays:** Perform midterm tests aiming to avoid possible unidentified errors in the preliminary solution.
9. **Depuration of the initial idea to avoid possible problems:** Process of problem-solving of the found errors in the preliminary concept (or prototype).
10. **Iteration of the depuration process:** Repeat the depuration process in order to ensure a good enough quality of the preliminary solution.
11. **Development of the final prototype:** Once all the problems have been solved, the final prototype will be designed and finally developed and assembled.
12. **Testing to ensure the functionality of the final solution:** Perform final tests to ensure the fulfilment of the requirements imposed on the solar panel solution.
13. **Study previous projects about attitude control systems applied to CubeSat:** Having into account that this part of the project will be focused on the improvement of an already developed device, an introductory study about previous projects will be needed.
14. **Study and analysis of previous control programs and hardware systems performed by previous students:** An introductory analysis of the programs developed by previous students will be also required to ensure a proper understanding of the spacecraft's functionality.
15. **Determination of an initial control algorithm:** Initial estimation of the control algorithm that is required to fulfill the control needing.
16. **Identification of possible hardware improvements:** Identification of some strategies to improve the original hardware device, such as MPU calibration or measuring by using DMP access.
17. **Depuration of the original code:** Correction of the malfunctions found in the preliminary control algorithm.
18. **Depuration of the hardware systems and new PCB redesign:** Correction of undetected hardware malfunctions and incompatibilities with the physical device.
19. **Hardware design and implementation of the proposed code:** Final assembly and connection of the hardware device and implementation of the already programmed code.
20. **Testing to ensure the proper behaviour of the system:** Test of the final electronic device to ensure the correct operation of the ACDS subsystem.
21. **Integration of solar array and attitude control systems:** Assembly of both the developed systems on a single 1U CubeSat prototype.
22. **Final functionality testing:** Final testing of the assembled prototype device to ensure proper behaviour of the final proposed solution.
23. **Final report writing:**Explanation and justification of the already defined tasks in the final output report.
24. **Discussion and conclusion:** Writing and justification of the conclusions once the thesis has been finished.
25. **Bibliography:** Writing of the bibliographic citations.

Number	**Task**	**Duration (weeks)**	**Predecessors**
1	Introduction	1	-
2	Searching and analysis of bibliography	1	1
3	Initial concept to be developed about solar arrays based on origami patterns	1	2
4	Development of the initial concept about attitude control applied to CubeSat technologies	1	2
5	Analysis of the solar array unfolding methods	0,5	3
6	Development of a solid concept about the proper solution applied to solar arrays	0,5	5
7	Design a preliminary prototype to determine the technical viability of the idea	2	6
8	Testing of the proposed solution applied to solar arrays	1	7
9	Depuration of the initial idea to avoid possible problems	0,5	8
10	Iteration of the depuration process	2	9
11	Development of the final prototype	3	10
12	Testing to ensure the functionality of the final solution	1	11
13	Study previous projects about attitude control systems applied to CubeSat	0,5	4
14	Study and analysis of previous control programs and hardware systems performed for previous students	1	13
15	Determination of an initial control algorithm	0,5	14
16	Identification of possible hardware improvements	0,5	15
17	Depuration of the original code	0,5	16
18	Depuration of the hardware systems and new PCB redesign	1,5	13
19	Hardware design and implementation of the proposed code	1	17/18
20	Testing to ensure the proper behaviour of the system	0,5	19
21	Integration of solar array and attitude control systems	3	12/2
22	Final functionality testing	1,5	21
23	Final report writing	4	22
24	Discussion and conclusion	1	23
25	Bibliography	0,5	23

Table 6.1 Order dependency table. Source: Own

6.1.2 Gantt diagram

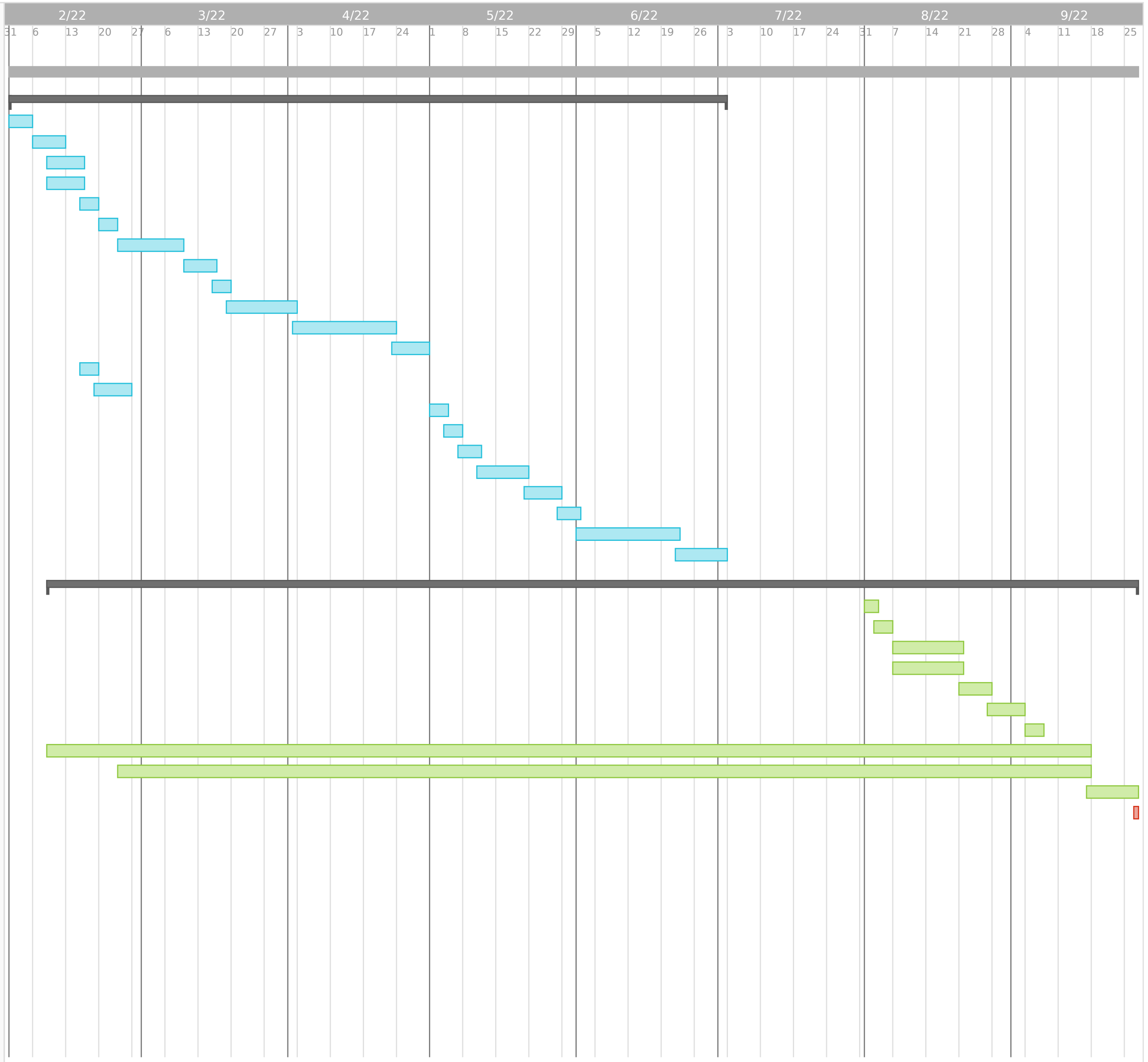
The following Gantt diagram is used to plan the project management and schedule. This tool allows showing the activities (i.e., tasks and events) displayed against time. On the left side of the chart, a list of the previously identified activities is displayed, while the horizontal axis shows a suitable and appropriate time scale with the maximum available time to end the project. Each activity is presented with its time allocation (see next page).

This allows to get an overview of the number of required activities, when they begin and how long each activity is scheduled to last, and also keeps a record of the project expected deadline.

Regarding the Gantt diagram that will be represented on the next page, the following considerations must be taken into account:

- Given that some of the tasks represented in the original time planning (represented in the project charter calendar) took more time than expected, a deadline extension was requested. Consequently, a redesign of the time planning had to be implemented in order to fit the new time requirements.
- In the final Gantt diagram, the task previously identified as Final report writing is represented in further detail and itemized in a group of smaller sub-tasks (represented in green in the diagram).

TFG_Gantt_diagram	start	end
Project development	**01/02/22**	**02/07/22**
Introduction	01/02	05/02
Bibliographic research	06/02	12/02
Deployable solar array initial concept	09/02	16/02
ADCS initial concept	09/02	16/02
Unfolding methods analysis	16/02	19/02
Solid concept about solar array solut...	20/02	23/02
Preliminary solar array prototype de...	24/02	09/03
Preliminary protoype tests	10/03	16/03
Depuration of preliminary protoype	16/03	19/03
Iteration of the deputation process	19/03	02/04
Development of the final prototype	02/04	23/04
Final prototype tests	23/04	30/04
Study of ADCS previous projects	16/02	19/02
Study of previously developed ADCS...	19/02	26/02
Determination of initial control algori...	01/05	04/05
Hardware improvements identificati...	04/05	07/05
Depuration of the original code	07/05	11/05
Depuration of hardware systems and...	11/05	21/05
Proposed code hardware implementa...	21/05	28/05
ADCS system tests	28/05	01/06
Integration of solar array and ADCS ...	01/06	22/06
Final functionality tests	22/06	02/07
Thesis Report Writing	**09/02/22**	**27/09/22**
Introduction	01/08	03/08
State of the art	03/08	06/08
Mechanics	07/08	21/08
Electronics	07/08	21/08
Integration of subsystems	21/08	27/08
Non-technical documents	27/08	03/09
Conclusions	04/09	07/09
Appendices	09/02	17/09
Blueprints	24/02	17/09
Report rebision	17/09	27/09
Final Submission	27/09	27/09



6.2 Budget

This section will be mainly dedicated to the analysis of the economic aspects of this project in terms of hardware, software, and human resources cost. To ensure the proper understanding of the cost analysis performed in these tables, this section has been structured according to the following breakdown:

6.2.1 Commercial components

During the execution of the project, several components were studied, justified, and bought, beginning from the original ADCS PCB components and going to the included in the deployable solar panel system, the manufactured pieces, and the normalized components used for their assembly.

In this subsection, all these mentioned divisions have been taken into account and represented in the following tables (Tables 6.2,6.3,6.4):

6.2.1.1 ADCS PCB

Ref	Description	Units	Unit cost (€/Unit)	Total Cost (€)
1	Breadboard	2	4,99 €	9,98 €
2	Cables	1	6,99 €	6,99 €
3	STM32 Blue Pill	1	13,49 €	13,49 €
4	USB ST-Link V2	1	10,27 €	10,27 €
5	GY-9250 IMU	1	8,23 €	8,23 €
6	SparkFun Qwiic Motor Driver	1	12,52 €	12,52 €
7	Qwiik cable	1	1,50 €	1,50 €
8	Motor Mabuchi RF-300EA-1D390	1	6,91 €	6,91 €
9	Bluetooth module HC-06	1	10,99 €	10,99 €
10	Air storage Kunshan Abama	1	200 €	200 €
Total				**280,88 €**

Table 6.2 ADCS PCB components budget. Source: [1]

6.2.1.2 Solar panel

Ref	Description	Supplier	Supplier Ref.	Units	Unit cost (€/Unit)	Total Cost (€)
11*	Stepper motor	ELEGOO	28byj-48 stepper motor	1	3,60 €	3,60 €
12*	Stepper motor driver	ELEGOO	ULN 2003 driver	1		
13	DC-DC Buck converter	JZK	MP1584	2	3,37 €	6,74 €
14	9V Battery	ENEGON	6F22	2	12 €	23,99 €
Total						**34,33 €**

Table 6.3 Solar panel components budget. Source: Own

***Note:** The components referenced as 11 and 12 corresponding to the stepper motor and its driver are bought in the same pack.

6.2.1.3 Normalized components

Ref	Description	Supplier	Norm	Units	Unit cost (€/Unit)	Total Cost (€)
11	Angular contact ball bearing	STK	7000AC	1	3,606 €	3,61 €
12	M3x16mm screw	DOLD Mechatronik	DIN 912	15	0,075 €	1,13 €
13	M3X8mm screw	DOLD Mechatronik	DIN 7991	12	0,164 €	1,97 €
14	M4X20mm	DOLD Mechatronik	DIN 7991	1	0,058 €	0,06 €
15	2x12mm pin	Integy Inc. (US)	DIN 6325	78	0,08 €	6,24 €
16	2x16mm pin	Integy Inc. (US)	DIN 6325	6	0,088 €	0,53 €
Total						**13,54 €**

Table 6.4 Normalized components budget. Source: Own

6.2.2 Additive manufacturing components

Regarding the fact that in this subsection the most closely related components to the development of the project will be economically studied, the table 6.5 will become the most detailed and complete of the whole budget.

In this table, the cost of the defined components will be exclusively determined by determining the weight of the consumed material during the 3D printing process of each piece. As a consequence, the following consideration must be taken into account:

- In order to also take into account the power consumption of the 3D printer device during its operation and considering that the determination of the precise number of work hours of the printer is almost impossible (given there are standby hours where the power consumption keeps going after the completion of the component printing), the manufacturing pieces cost associated with the consumed material will be multiplied by a factor of 2,5 in the final budget.

Ref	**Description**	**Project.Ref**	**Units**	**Weight (g)**	**Unit cost (€/Unit)**	**Total Cost (€)**
1	ADCS PCB	1.03	1	-	280,88 €	280,88 €
2	Reaction wheel	1.04	1x2	20,73	0,4973 €	0,9946 €
3	Origami support fixation	1.05	1x2	1,00	0,0240 €	0,0480 €
4	Origami platform	1.06	1x2	17,32	0,4155 €	0,8301 €
5	Origami counter-platform	1.07	1x2	1,78	0,0427 €	0,0854 €
6	Air bearing fixation	1.08	4	16,28	0,3905 €	1,5622 €
7	Battery + DC converter support	1.01.02	1x3	10,75	0,2579 €	0,7737 €
8	CubeSat base wall	1.01.03	1x2	31,84	0,7638 €	1,5277 €
9	CubeSat side wall	1.01.04	2x2	27,39	0,6571 €	2,6283 €
10	CubeSat side wall + PCB	1.01.05	2x2	28,49	0,6835 €	2,7339 €
11	CubeSat superior wall	1.01.01.01	1x2	37,05	0,8888 €	1,7776 €
12	Battery support	1.01.01.02	1x3	5,43	0,1303 €	0,3908 €
13	Driver + DC converter support	1.01.01.03	1x2	6,88	0,1651 €	0,3301 €
14	Pinion gear	1.01.01.04	1x9	1,46	0,0350 €	0,3152 €
15	Deployable system toothed ring	1.02.01.02	1x2	7,25	0,1739 €	0,3479 €
16	Deployable system base hexagon	1.02.01.03	1x2	18,88	0,4529 €	0,9059 €
17	Deployable system top hexagon	1.02.01.04	1x2	18,84	0,4520 €	0,9039 €
18	V-shape hinge	1.02.02.01	6x2	0,71	0,0170 €	0,2044 €
19	V-shape tight rod	1.02.02.02	12x2	0,07	0,0017 €	0,0403 €
20	V-shape sliding rod	1.02.02.03	6x2	2,25	0,0540 €	0,6477 €
21	V-shape fixation ring	1.02.02.04	6x2	2,00	0,0480 €	0,5758 €
22	Deployable arm sliding rod	1.02.01.01.01	24x2	0,79	0,0190 €	0,9097 €
23	Deployable arm tight rod	1.02.01.01.02	48x2	0,90	0,0216 €	2,0727 €
24	Deployable arm sliding half rod	1.02.01.01.03	6x2	0,42	0,0101 €	0,1209 €
25	Deployable arm tight half rod	1.02.01.01.04	12x2	0,48	0,0115 €	0,2764 €
	Total printed pieces					**21,00 €**
	Total					**301,88 €**

Table 6.5 Additive manufacturing components budget. Source: Own

6.2.3 Software licenses

In the following budget table 6.6 the cost of the used software tools has been represented. Nevertheless, given that this is a completely academic project, and the Polytechnical University of Catalonia owns educational licenses for all these programs, the costs represented in this subsection could not be considered in the particular scenario at which this project has been developed.

However, these considerations must be taken into account by external investigators or companies, and due to this requirement, the costs of this contribution will be also represented in the final budget table 6.8.

Ref	**Software**	**Units**	**Cost (€)**
1	Solidworks	1	1295 €
1	Matlab	1	800 €
1	Arduino IDE	1	0 €
1	KiCad	1	30,40 €
1	Fritzing	1	8 €
Total			**2133,40 €**

Table 6.6 Software licences budget. Source: Own

6.2.4 Human resources

According to the Idescat (Statistics Institute of Catalonia), the mean hourly salary for a professional investigator, scientist, or engineer [35] is around 25.28 €/h; and the time invested in this research project was about 800 hours (considering it is a double-degree final project). As a consequence, the following manpower costs must be accounted:

Ref	**Description**	**Hours of development**	**Cost/Hour (€/Hour)**	**Total Cost (€)**
1	Human resources for project development	800	25,28 €	**20224 €**

Table 6.7 Human resources budget. Source: Own

According to table 6.7, human resources costs will represent the major part of the total estimated costs for this project.

6.2.5 Final budget

Once all the previously broken down costs have been justified, the final cost estimation can be calculated as follows (Table 6.8).

Ref	**Description**	**Cost (€)**
1	ADCS PCB	**280,88 €**
1	Solar panel	**34,33 €**
1	Normalized components	**13,54 €**
1	Additive manufacturing components	$\mathbf{21 \cdot 2,5 =}$ **52,50€**
1	Software licences	**2133,40 €**
1	Human resources	**20224 €**
Total		**22738,65 €**

Table 6.8 Final budget. Source: Own

6.3 Environmental impact

In this section, the environmental impact of the project will be estimated and analyzed. Despite the major advances in the aerospace area, there are still secondary effects that need to be taken into account and are not directly related to the development of these technologies.

Sustainability is defined as the capacity of preserving an activity within a long period of time without harming the environment. However, the most accepted definition of sustainability is provided by the World Commission on Environment and Development Definition.

The major dilemma of satellites launching into orbit resides in the space debris it produces. At the same time, one of the major concerns among scientists around the world is not only the impact it might have on other exploration missions but also the light pollution it has in the celestial vault.

Any project that implies a satellite shall analyze the effect among other outer space activities. For instance, the accumulation of space debris can lead to a chain reaction that can cause severe damage to other missions. In spite of this fact, the carbon footprint of a spacecraft once launched is nearly zero, since it has solar panels that will supply the required energy.

Since the scope of this project is focused entirely on academic and development aspects, and this development has been carried out in laboratory conditions, there are no more environmental issues than the estimated CO2 emission during the analysis and development of it (See table 6.9).

$$m_{CO_2} = Impactfactor(\frac{kgCo_2}{KW \cdot h}) \cdot Energy(KW \cdot h) \tag{6.1}$$

According to 6.1 expression, the values in table 6.9 has been calculated.

Activity	**Time (h)**	**Power (KW)**	**Impact factor** ($kgCo_2/kg \cdot h$)	**kg** CO_2
Laptop	500	0,3	0,2	**30**
3D manufacturing	300	0,12	0,2	**7,2**
Total				**37,2 kg**

Table 6.9 Power consumption and CO_2 emissions. Source: Own

Chapter 7

Conclusions

To sum up, the present thesis had the main objective of developing new devices for CubeSat applications in order to improve both the mechanical and electronic performance of a nanosatellite prototype for testing purposes. All these contributions were encompassed in the manufacturing of two different subsystems, an Origami-based deployable solar panel subsystem, and the improvement of the ADCS device (based on a previously studied attitude control system, which has been debugged and enhanced with new functionalities).

When talking about the Origami-based deployable solar panel system, it can be concluded that, regarding the mechanical simulated tests, the geometrical considerations during its development and design, the design iteration in order to avoid malfunctions, and the final operation test which lead to outstanding results; the final result has more than fulfilled the initial expectations of this device.

Regarding the ADCS developed system, it can be summarized that, considering the implemented improvements in the measurement devices by acceding the DMP, the easy calibration method achieved, the improvements over the attitude control algorithm, and the results of the tests performed to ensure its correct behavior; the final achievement of the imposed control requirements is completely guaranteed. It must also be mentioned that the integration of the automatic deployment of the solar panel system has also met the imposed expectations.

To give a global conclusion of the whole project, and regarding the previous paragraphs, it can be brought to an end that the objective of the thesis is accomplished and properly justified during the execution of this final report.

7.1 Summary of contributions

So as to clarify and better exemplify the overall contributions to the aerospace field that this project has achieved, the following table is presented (Table 7.1):

Engineering fields	Contributions
Mechanical	- Origami-based solar panel design and manufacturing - Deployment mechanism design and manufacturing - Reaction wheel redesign - ADCS PCB hardware integration - Design and manufacturing of CubeSat frame - Power supply system hardware integration - Adaptation of the prototype to the air-bearing testbed
Electronic	- MPU measurement by DMP access - Calibration method improvement - Solar panel deployment code integration - Deployment control hardware integration - ADCS PCB redesign

Table 7.1 Table of contributions. Own

7.2 Future works

The main objective of this section is to enumerate, introduce and justify some aspects of the developed project that could be further improved and the strategies that could be followed in order to achieve these improvements.

- Considering that the Origami-based solar panels are not implemented in the developed prototype, some further improvements can be achieved if real flexible solar panels are attached, connected, and tested at the designed structure. By studying the power supply performance of these real solar arrays, deeper knowledge about the viability of this technology could be researched.
- Regarding the complex structural movement that the proposed origami pattern presents, a further study of the movement of its vertexes and edges could be required to adapt these structures to the aerospace industry. To do that, a deeper study of this pattern could be achieved by using computational simulation tools, such as Rhinoceros-Grasshopper software offers.
- Taking into account some new-born auto-deployable technologies that have been developed during the lasts years, such as the development by Oxford Space Systems, these technologies could be also implemented to deployable solar array subsystems, such as the one studied in this project, to further enhance its performance.
- According to the material selection justified in this thesis, the manufacturing of a prototype by using plastic materials has some limitations in terms of device integrity tests in real conditions. So as to get a deeper reference of this behavior, the prototyping of a device by using real aerospace-application materials could be required.

- When talking about Attitude Determination and Control Subsystems, some other control actuators, apart from reaction wheels, are implemented, such as magnetorquers. Therefore, the attitude control capabilities of the device could be further improved by implementing these actuators to produce additional functions, such as detumbling stages.
- To further improve the ADCS capabilities of the developed prototype, the proper implementation, and tuning of the Fine pointing mode cascade control loop is of paramount importance.
- According to Iván Gonzálvez's [36] and Joan Serrano's [33] thesis, 3DoF control can be implemented by adding some deeper and more accurate control algorithms, aside from implementing 3 reaction wheels. By doing so, complete control over the positioning of a spacecraft could be achieved.
- Because of some limitations in the DC motor used for the ADCS subsystem, proper control of the electrical variables of the motor can only be achieved by designing a much more complex PCB circuit. So as to avoid this complexity, the implementation of brushless motors could significantly enhance the control performance.
- The implementation of a graphical interface to easily read and compare the measured results, and, at the same time, configure the setting of the executed tests, could be a great help for all researchers who would use this simulation device.

Bibliography

[1] Yi. Qiang, Project of 1 DoF Attitude Control System of 1U CubeSat Based On Reaction Wheel. Terrassa, Spain: UPC, 2021.

[2] Adrià. Pérez, Project of attitude control subsystem based on reaction wheels. Terrassa, Spain: UPC, 2021.

[3] David. Gonzalez, Integrated Hardware in the loop simulation PLATform of Optical communication in Nanosatellites. Terrassa, Spain: UPC, 2021.

[4] California Polytechnic State University. "CubeSat Design Specification." https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf (accessed Feb. 26, 2022).

[5] ISISPACE GROUP. "CubeSat information." https://www.isispace.nl/cubesat-information (accessed Feb. 26, 2022).

[6] Canadian Space Agency. "Activities sectors: CubeSat." http://www.asc-csa.gc.ca/eng/satellites/cubesat/what-is-a-cubesat.asp (accessed Mar. 16, 2022).

[7] ISISPACE GROUP. "ISIPOD CubeSat Deployer." https://www.isispace.nl/product/isipod-cubesat-deployer/ (accessed Mar. 24, 2022).

[8] NASA. "Starshade Technology Development." https://exoplanets.nasa.gov/exep/technology/starshade/ (accessed Feb. 15, 2022).

[9] F. Escrig, "Expandable space structures," International Journal of Space Structures, Vol.1, No.2, pp.79–91, 1985.

[10] S.D. Guest and S. Pelegrino, "A new concept for solid surface deployable antennas," Acta Astronautica, Vol.38, No.2, pp.103–113, 1996.

[11] Duan. Baoyan, "Large Spaceborne Deployable Antennas (LSDAs) - A Comprehensive Summary," Chinese Journal of Electronics Vol.29, No.1, 2020.

[12] Rongqiang Liu, DakeTian and Zongquan Deng, "Research status and prospects of spaceborne deployable antenna structure," Mechanical Design, Vol.27, No.9, pp.1–21, 2010.

[13] Tuanjie Li and Xiaofei Ma, "Large spaceborne expandable antenna technology," Space Electronics, Vol.9, No.3, pp.35–43, 2012.

[14] Rory Barrett, Douglas Campbell, Development of a Passively Deployed Roll-Out Solar Array. Defense Technical Information Center, 2006.

[15] NASA. "SpaceX CRS-11 Mission Overview." https://www.nasa.gov/sites/default/files/atoms/files/spacex_crs-11_mission_overview.pdf (accessed Aug. 6, 2022).

[16] Marc. Garcia. “New Solar Arrays to Power NASA’s International Space Station Research.” https://www.nasa.gov/feature/new-solar-arrays-to-power-nasa-s-international-space-station-research (accessed Aug. 6, 2022).

[17] NASA, Current and Future Operations and Challenges with International Space Station. ISS Program Office, 2020.

[18] S. Chesi, “A Dynamic, Hardware-in-the-Loop, Three-Axis Simulator of Spacecraft Attitude Maneuvering with Nanosatellite Dimensions,” Journal Of Small Satellites, vol. 4, nº 1, pp. 315-328, 2015.

[19] Arend L. Schwab and Jaap P. Meijaard, “How to draw Euler angles and utilize Euler parameters,” https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.723.1234&rep=rep1&type=pdf International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol.42568, pp.259–265, 2006.

[20] ISISPACE GROUP. “ISIS Magnetorquer Board.” https://www.isispace.nl/product/isis-magnetorquer-board/ (accessed Mar. 13, 2022).

[21] Zuliana Ismail and Renuganth Varatharajoo. “A study of reaction wheel configurations for a 3-axis satellite attitude control.” https://www.sciencedirect.com/science/article/pii/S0273117709007078 (accessed Mar. 6, 2022).

[22] EnduroSat. “1U CubeSat Solar Panels X/Y.” https://www.endurosat.com/products/cubesat-solar-panels-x-y/ (accessed Feb. 19, 2022).

[23] Omar. Fabrisio, Diseño paramétrico de la estructuras desplegables. Granada, Spain: Departamento de Tecnología de la Arquitectura. Escuela Técnica Superior de Arquitectura del Valles, ETSAV, 2020.

[24] NASA. “Modular Frame Designs.” https://www.nasa.gov/smallsat-institute/sst-soa/structures-materials-and-mechanisms (accessed Apr. 16, 2022).

[25] Antoni. Bonilla, Estudio, Diseño y construcción de una plataforma pivotante para ensayo del sistema de control de actitud (ACS) de un cubesat. Terrassa, Spain: UPC, 2019.

[26] Roger B. Nelsen, “Proof Without Words: The Area of a Regular Dodecagon,” The College Mathematics Journal, vol. 46, nº 1, 2015.

[27] Martin I. Hoffert and Ken Caldeira. “Climate change and energy, overview.” https://www.sciencedirect.com/science/article/pii/B012176480X00396X (accessed Aug. 7, 2022).

[28] Junlan. Li. “Thermal analysis of composite solar array subjected to space heat flux.” https://www.researchgate.net/publication/260988045_Thermal_analysis_of_composite_solar_array_subjected_to_space_heat_flux (accessed Aug. 12, 2022).

[29] Pilar M Espinet-Gonzalez, Enrique J Barrigón, Gaute S Otnes, Giuliano Vescovi, Colin Mann, Ryan France, Alex Welch, and Matthew Hunt, https://pubs-acs-org.vu-nl.idm.oclc.org/doi/10.1021/acsnano.9b05213 (accessed Aug. 18, 2022).

[30] Juan. Palomares, Estudio y diseño de dos placas de intercambio de datos de inclinación y posición entre dos CubeSat. Terrassa, Spain: UPC, 2019.

[31] Evan G. Hemingway and Oliver M. O’Reilly, “Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments,” Multibody System Dynamics, vol. 44, nº 1, pp.31–56, 2018.

[32] Charlote. Treffers and Luc. van Wietmarschen, Position and orientation determination of a probe with use of the IMU MPU9250 and a ATmega328 microcontroller, 2016.

[33] Joan. Serrano, Attitude control techniques for a 1U CubeSat. Terrassa, Spain: UPC, 2022.

[34] Jordan. Morales, Desarrollo de una placa de control de actitud 2D para un prototipo de CubeSat con pares magnéticos y ruedas de inercia. Terrassa , Spain: UPC, 2021.

[35] Idescat. "Anuario estad´ıstico de Catalu˜na. Salario bruto anual y ganancia por hora. Por sexo y tipo de empleo." https://www.idescat.cat/pub/?id=aec&n=398&lang=es (accessed Sept. 5, 2022).

[36] Iván. Gonzálvez, Modeling, sizing and simulation of the power subsystem of CubeSat with Matlab/Simulink. Terrassa, Spain: UPC, 2022.

[37] Michael Douglas. Griffin, Space vehicle design. AIAA, 2004.

[38] European Cooperation for Space Standardization. "ECSS-E-ST-10C Rev.1 – System engineering general requirements (15 February 2017)." https://ecss.nl/standard/ecss-e-st-10c-rev-1-system-engineering-general-requirements-15-february-2017/ (accessed Sept. 8, 2022).

[39] NASA. "NASA Systems Engineering Handbook Revision 2." https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook (accessed Sept. 8, 2022).

[40] SEBoK. "Guide to the Systems Engineering Body of Knowledge (SEBoK)." https://www.sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK) (accessed Sept. 12, 2022).

[41] ISO. "ISO - ISO/IEC/IEEE 15288:2015 - Systems and software engineering — System life cycle processes." https://www.iso.org/standard/63711.html (accessed Sept. 2, 2022).

[42] Air Force Space Command. "Range Safety User Requirements Manual Volume 3." https://static.e-publishing.af.mil/production/1/afspc/publication/afspcman91-710v3/afspcman91-710v3.pdf (accessed Sept. 15, 2022).

Appendix A

Deployable solar panel subsystem conception process

Since the selection of a suitable approach to reach the desired solution is of crucial importance for the proper development of any new prototype device, this annex will be exclusively dedicated to the explanation of the thinking process carried out to give birth to the deployment mechanism for the Origami-based solar panel final solution concept, properly justified during the main report of this project (Chapter 3).

So as to properly represent this iterative brainstorming process, this annex will independently examine the deployable structures and the torque transmission mechanisms:

A.1 Deployable arms-based candidates

To determine the optimum solution for the rhomboidal shape deployable arm configuration, two different assembly structures have been studied:

A.1.1 Single layer deployable arm

This arm configuration is based on a single rhomboidal-shaped deployable layer in which rods are connected by using steel pins and taking advantage of the dimensional tolerance of the rods and the elastic nature of polymeric materials to still allow movement. The geometry of this structure is represented in figure A.1.

Some of the advantages of this configuration are:

- **Structural simplicity:** The single-layer rhomboidal structure allows an easy assembly of the arm configuration as well as helps in the final integration with the torque transmission mechanism.
- **Low weight:** The low amount of pieces of this configuration also optimize its payload for aerospace applications.
- **Deployment simplicity:** The easy unfolding method of this typology of deployable arms helps to the simplification of the final torque transmission mechanism.

Despite these advantages, the use of this arm configuration could lead to potential technical issues when unfolding the Origami-shaped solar panel due to its lack of stiffness. Consequently, the following arm structure of higher complexity have also been studied.

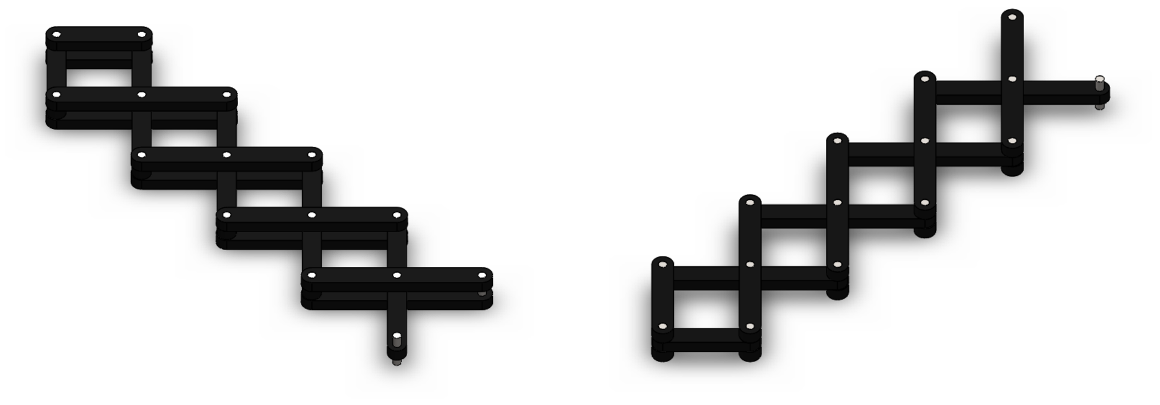

(a) Single layer deployable arm isometric back view. **(b)** Single layer deployable arm isometric front view.

Figure A.1 Single layer deployable arm 3D model assembly. Source: Own

A.1.2 Multiple layer deployable arm

So as to give a solution to the lack of stiffness that the previous arm configuration could present, a three-layer rhomboidal deployable arm structure has been studied. This arm typology is based on the assembly of three rhomboidal arms, like the ones already explained, which form an H-like side shape. Consequently, the connection of the three independent arms to produce the desired mechanism requires the design of new pieces, which can be seen at figure A.2.

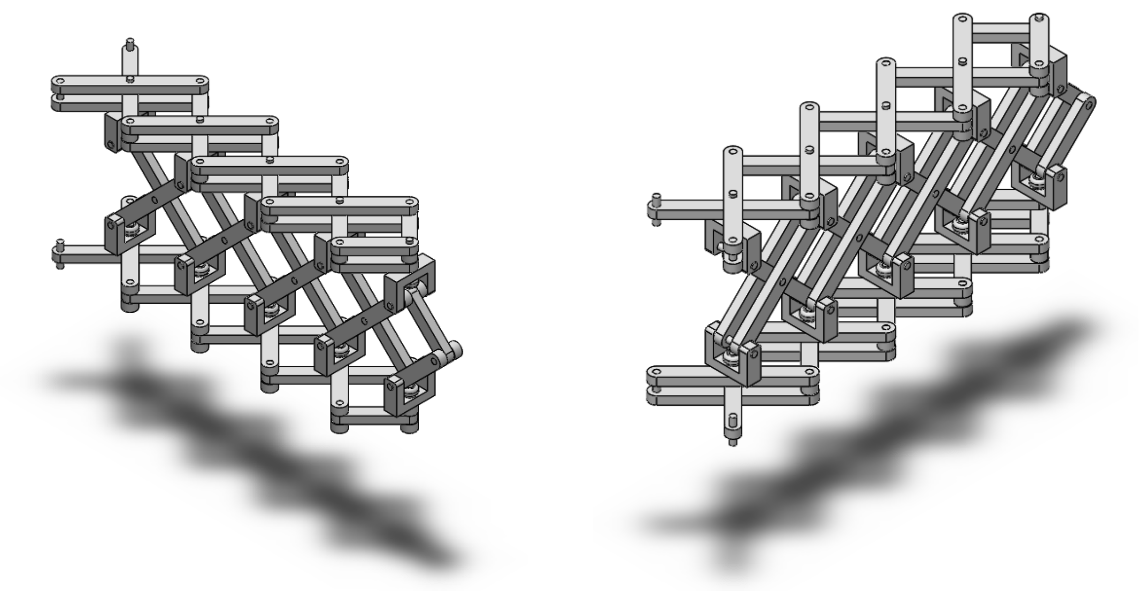

(a) Triple layer deployable arm front view. **(b)** Triple layer deployable arm back view.

Figure A.2 Triple layer deployable arm 3D model assembly. Source: Own

Despite the fact that this configuration enhances the stiffness of the whole deployment subsystem, the payload limitations and the great complexity that this structure implies would result in a non-optimum solution for the final purpose. Consequently, the finally selected arm configuration is the single layer deployable arm concept.

A.2 Torque transmission mechanism candidates

Once the configuration of the deployable arms has been selected, it is now time to build up a solid concept for the torque transmission mechanism, which will be in charge of the unfolding of the whole structure, also including the previously seen deployable arms. In order to do so, two different approaches have been considered:

A.2.1 Self contraction deployment mechanism

Regarding the already selected arm configuration, this deployable mechanism is based on the unfolding of the single-layer rhomboidal arms by applying a contraction force between the tips of the arm rods.

So as to do so, the unfolding device will present two hexagonal pieces, which are placed in parallel, and by applying enough force to bring them closer generate the unfolding of the whole structure. Figure A.3 represents the basic configuration of this mechanism:

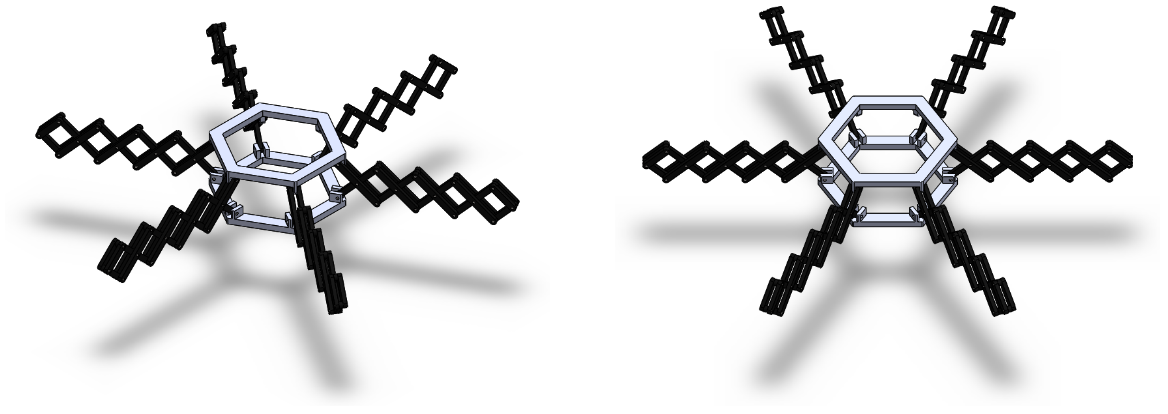

(a) Self contraction deployment mechanism side view. **(b)** Self contraction deployment mechanism top view.

Figure A.3 Self contraction deployment mechanism 3D model assembly. Source: Own

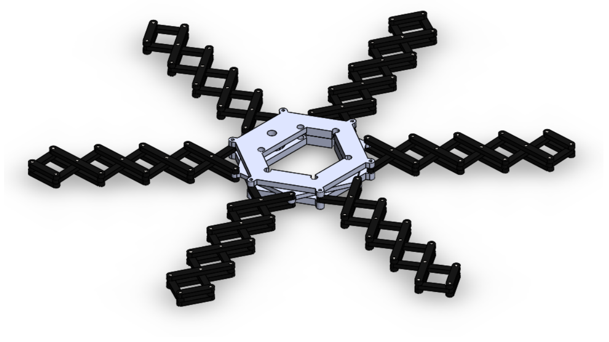
Regarding the fact that, to generate the already mentioned contraction force between the two hexagonal pieces a cable system would be required, and the high complexity that these mechanisms imply, the implementation of this idea in the final prototype would lead to some notable problems. It must also be mentioned that cable systems tend to be voluminous, and consequently, would not fit the needed space requirements.

A.2.2 Single gear counter-rotation deployment mechanism

According to the selected arm configuration and the unfolding method by bringing closer the rod's tips considered in the previous mechanism, an alternative torque transmission method has been studied.

In order to avoid the use of cable systems, a gear-based mechanism has been considered (Figure A.4a).

The basic operating principle of this alternative is the generation of a counter-rotation movement between two hexagonal pieces which, by attaching the rod's tips to its vertexes, would lead to the unfolding of the rhomboidal arms and, consequently, the deployment of the whole system (Figure A.4b).


(a) Single gear deployment assembly.

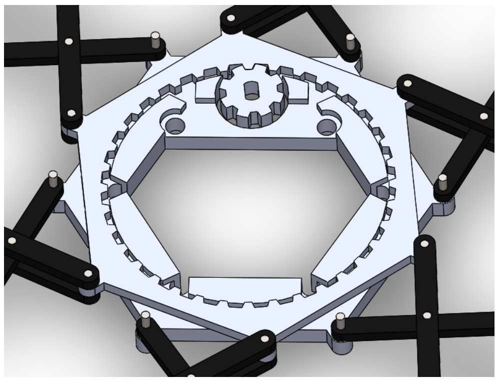
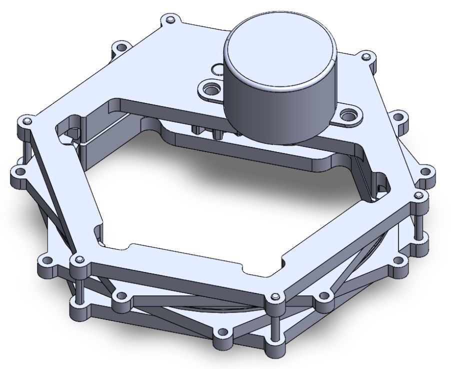

(b) Single gear deployment mechanism.

Figure A.4 Single gear counter-rotation deployment mechanism 3D model assembly. Source: Own

A.2.3 Multiple gear counter-rotation deployment mechanism

Taking a further step on the concept of the previously seen gear-based mechanism, and with the objective to avoid the production of a parasite torque associated to the rotation of one of the hexagonal pieces while maintaining the rest of the structure still, a more complex multiple gears-based mechanism has been evaluated.

Despite the fact that this mechanism is very similar to the previously mentioned gear-based one, its main difference lies in the rotation movement of the rod's tips attached to the hexagonal pieces (Figure A.5a). While in the single-gear mechanism only one of the rod's tips is rotated, in this multiple-gear mechanism both tips are rotated at the same time in opposite rotation directions. This double rotation of the rod's tips is achieved by adding a second pinion gear to the mechanism, which inverts the rotation direction and transmit it to a second moving hexagonal piece (Figure A.5b).


(a) Multiple gear deployment assembly.

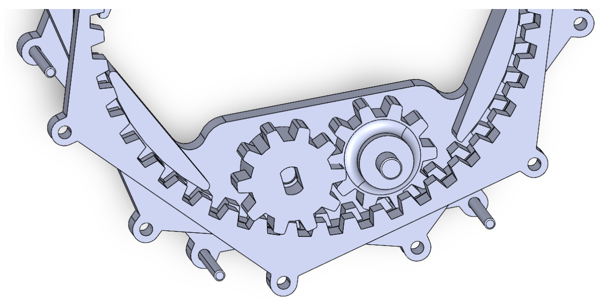

(b) Multiple gear deployment mechanism.

Figure A.5 Multiple gear counter-rotation deployment mechanism 3D model. Source: Own

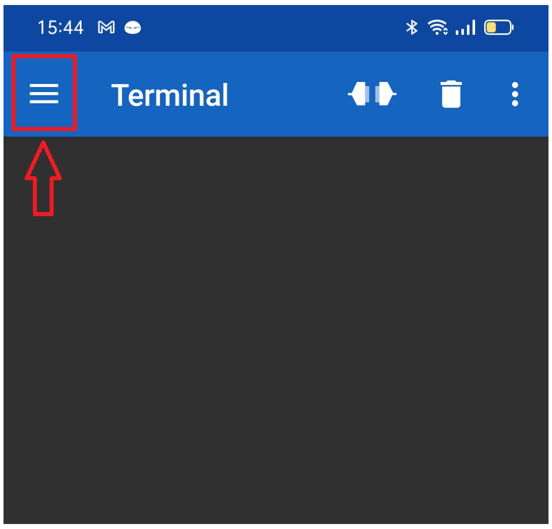
As a consequence of the great complexity that this modification implies in the final system, and according to the payload and volume limitations, this alternative is finally discarded. Therefore, the final implemented deployable mechanism solution will be the single-gear counter-rotation deployable mechanism (Figure A.4a).

Appendix B

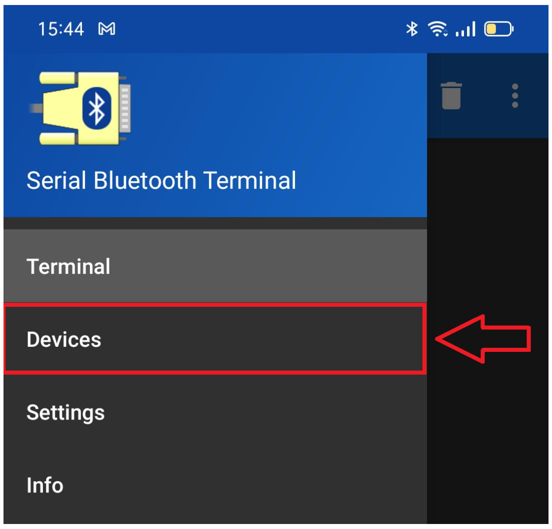
Prototype operation guidelines

If the tests performed during this project are to be reproduced, the following considerations about the used lab equipment and the data recording method must be taken into account:

1. Firstly, the mobile application "Bluetooth Terminal" must be downloaded from "Google Play".
2. Install the application on the mobile phone.
3. Enable Bluetooth communication on the mobile device.
4. Open the Bluetooth Terminal application.
5. Click Options (Figure B.1a) > Devices (Figure B.1b) > Select "HC06_Plathon" device.



(a) Options selection button.



(b) Device selection.

Figure B.1 Bluetooth connection process. Source: Own

6. Introduce the password "1234" to enable communication.

7. Press the connection button on the main screen of the application (Figure B.2).

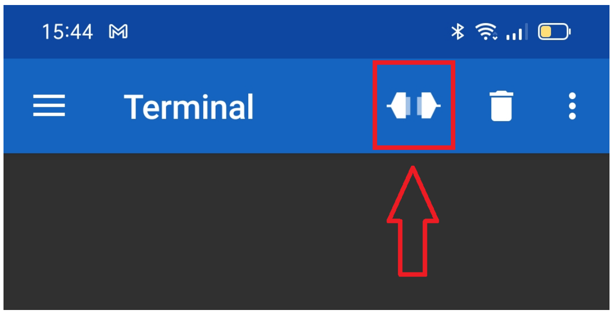


Figure B.2 Connection button. Source: Own

8. Once a connection message "Connected" is displayed on the screen, press the reset button on the STM32 Bluepill microcontroller to execute the coded program.
9. Place the prototype on the air-bearing platform.
10. To open the air flux, rotate the handle in the clockwise direction (Figure B.3). The flowmeter shown in Figure B.3 must be fixed between 10lpm and 15lpm.

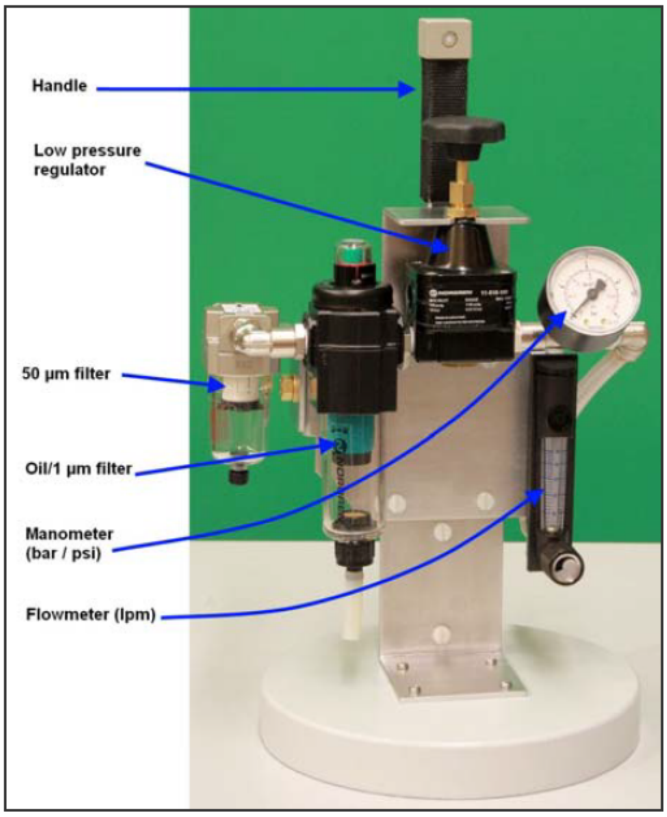


Figure B.3 Air conditioning unit. Source: Appendix F.6

11. To operate the prototype device, the operating mode must be firstly selected introducing 1,2,3, or 4 using the keypad.
12. Finally, depending on the selected operating mode, the rotation set point can be introduced (only for mode 1), where the rotation direction will be fixed for the sign of the introduced value ('+' for counterclockwise rotations, and '-' for clockwise rotations).

Appendix C

Requirements

C.1 Technical requirements

In this appendix, strict and measurable statements will be provided to guide and limit the process, as well as to avoid weak points and vague actions that could lead to misunderstandings:

Regarding the main goal of this project, which is the design and manufacturing of an ADCS prototype for testing purposes in laboratory conditions, the final proposed solution must meet the following technical requirements:

- According to The CubeSat standard specification [4], the maximum dimensions of 1U CubeSat must not exceed a 10x10cm side cube with a maximum weight of 1,3kg.
- So as to be categorized as a nanosatellite, its weight must be higher than 1kg and less than 10kg.
- According to Dr. Griffin and Dr. French [37], the ADCS subsystem weight is around 10-15% of the total payload of the satellite. This implies that its ADCS subsystem must not exceed 200g weight.

Nevertheless, if the developed ADCS and solar panel system is going to be adapted for real launching applications, the following formal document-based and model-based standards should also be met:

- European Cooperation for Space Standardization (ECSS): ECSS-E-ST-10C [38]
- The National Aeronautic and Space Administration (NASA): Systems Engineering Handbook [39].
- International Council on Systems Engineering (INCOSE): Gide to the Systems Engineering Body of Knowledge (SEBoK) [40].
- ISO/IEC/IEEE 15288: System and software engineering [41].

C.2 Structure subsystems

CubeSat specifications can be found in California University's specifications [4]. However, this subsection will be dedicated to the General, Mechanical, and Electrical requirements of the developed device, also including some requirements to take into account during the development to be applied in real launching missions.

C.2.1 General requirements

1. All parts shall remain attached to the CubeSat during launching, ejection and operation. To ensure this requirement, the CubeSat designed frame must support the forces it is subjected.
2. No pyrotechnics shall be permitted.
3. Any propulsion systems shall be designed, integrated, and tested in accordance with AFSPCMAN 91-710 Volume 3 [42].
4. Total stored chemical energy will not exceed 100 W·h.
5. CubeSat hazardous materials shall conform to AFSPCMAN 91-710, Volume 3 [42].

C.2.2 Mechanical requirements

1. Deployables shall be constrained by the CubeSat, not the P-POD.
2. The maximum mass of a 1U CubeSat shall be 1,33kg.
3. The maximum mass of a 1,5U CubeSat shall be 2,00kg.
4. The CubeSat center of gravity shall be located within 2cm from its geometric center on the X and Y axis.
    - The 1U CubeSat center of gravity shall be located within 2 cm from its geometric center in the Z direction.
    - The 1,5U CubeSat center of gravity shall be located within 3 cm from its geometric center in the Z direction.
5. Aluminum 7075, 6061, 5005 and/or 5052 will be used for the main CubeSat structure (For real CubeSat development).
6. ADCS PCB shall be properly cushioned so as to prevent vibrations that could lead to measurement issues.

C.2.3 Electrical requirements

1. The CubeSat power system shall be at a power-off state to prevent CubeSat from activating any power functions when it is not being used for testing purposes.
2. The power supplier devices shall be 9V batteries so as to fulfill the devices' power requirements.

C.3 Operational requirements

This CubeSat prototype will meet certain requirements regarding the integration and operation of the device.

1. Prior to the operation of the device, the deployable mechanism must be properly lubricated to ensure the correct unfolding of the system.
2. Prior to any test, the connections of both the ADCS system and stepper motor applied to the deployable mechanism must be tested and visually inspected.
3. When the external conditions are significantly changed, and prior to any IMU measurement testing, the MPU must be properly calibrated using the method specified in Chapter 4.
4. When placing the developed prototype in the air-bearing device, the balance of the CubeSat prototype must be visually verified to prevent unexpected movements while testing.
5. When the deployable system is being folded, a visual inspection must be performed to prevent the collision of the deployable arms with the cube's wall, as a consequence of the displacement of the arm's movement axis by action of its own weight when fully extended.

Appendix D

Developed and used codes

D.1 ADCD system Arduino code

```
/* Reaction Wheel controller
 *
 * The following code provides a control for PLATHON project's CubeSat.
 *
 * The following libraries are used:
 * STM32 Bluepill microcontroller package from http:/ ...
    /dan.drown.org/stm32duino/package_STM32duino_index.json
 * SCMD Driver library by SparkFun used (https:// github.com/sparkfun/
 SparkFun_Serial1_Controlled_Motor_Driver_Arduino_Library)
 * "MPU6050_6Axis_MotionApps20.h" library by Juan Palomares
 * read_IMU() and IMU setup content code extracted from Andres Gomez ...
    and Miquel Reurer, from the PLATHON group (magnetorquers section)
 */

// _________________________________________________________________ ...
   DEFINITIONS
// _________________________________________________________________ ...
   LIBRARIES
#include <Arduino.h>     // Arduino library
#include <stdint.h>      // Integer types library
#include <SCMD.h>        // Serial Controlled Motor Driver library
#include <SCMD_config.h> // Serial Controlled Motor Driver ...
   Configuration library
#include <Wire.h>        // I2C library
//IMU librarys
#include "MPU6050_6Axis_MotionApps20.h" //Mediante esta biblioteca se ...
   pueden obtener datos del DMP (Digital Motion Processor) de la IMU
#include "I2Cdev.h" //Mediante esta biblioteca se pueden obtener ...
   datos del DMP de la IMU
#include "helper_3dmath.h" //Mediante esta biblioteca se pueden ...
   realizar operaciones de cuaterniones

```

```

// __________________________________________________DEFINITIONS
#define LEDPIN PC13                   // Integrated LED of the Bluepill
#define STM32_CLOCK 72000             // Internal STM32 Clock (72MHz, in ...
    kHz so period is millisec)
#define PI 3.14159265                 // Number PI
#define Rad_to_deg 57.29577951        // Convert radians to degrees
#define Deg_to_rad 0.01745329         // Convert degrees to radians
#define Final_pointing_tolerance 1 // Tolerance for final pointing ...
    (+-1 degree of tolerance) (Check if position is within the margin)
#define Pointing_mode_tolerance 5  // Tolerance for selecting ...
    pointing mode (+-5 degree of tolerance) (Select whether to use ...
    coarse or fine mode)
#define Accel_tolerance 1.5           // Tolerance for acceleration ...
    measurement (+-0.5 unit of acceleration of tolerance) (Acceptable ...
    error of acceleration)
#define Gyro_tolerance 3              // Tolerance for gyro measurement ...
    (+-1 unit of gyroscope Z tolerance) (Accounts to know if it is ...
    rotating or not at a constant speed for the ramp)

SCMD DriverOne;           // Driver Object definition
//MPU9250 IMU(SPI, CS1); // MPU object definition (Not used in this code)

//__________________________________________________STEPPER MOTOR VARIABLE ...
    DEFINITION
#define IN1  PB15
#define IN2  PB14
#define IN3  PB13
#define IN4  PB12
bool pulse_stop = false;
//int speedM= 12e12;
int speedM= 20;

// Step sequence definition (maximum torque)

int step_open [4][4] =
{
  {1, 0, 0, 1},
  {0, 0, 1, 1},
  {0, 1, 1, 0},
  {1, 1, 0, 0}
};


int step_close [4][4] =
{
  {1, 1, 0, 0},
  {0, 1, 1, 0},
  {0, 0, 1, 1},
  {1, 0, 0, 1}
};
```

```

//COARSE mode variable declaration and initialization
   bool state_zero_CW=false, ...
      state_zero_CCW=false,CBS_direction=false, not_accel=false;
   float ∆_degree, initial_degree, final_degree, Yaw_deg_prev = 0;
   bool Pointing = false, End_Coarse= false;
   float initial_speed=20, impulse_speed=150, final_speed=20, ...
      decelering_speed=80;

//__________________________________________________________________IMU
const int mpuAddress = 0x68;  // The IMU adress is defined, it can be ...
    both 0x68 or 0x69
MPU6050 mpu(mpuAddress); //An object MPU6050 is created, pulse_stop ...
    extracts data form the DMP
int fifoCount = 0, packetSize; //A FIFO value counter is created and ...
    the FIFO packet size is defined
byte fifoBuffer[42]; //A buffer is created, which will be used for ...
    pulse_stop to obtain the roll, pitch and yaw data.
float roll, pitch, yaw, sqx, sqy, sqz, sqw, test; //Roll, pitch y yaw ...
    are defined as well as the needed "float" variables
Quaternion q; //A quaterinon is defined to obtain the roll, pitch and ...
    yaw values.

// ...
    ___________________________________________________________________________ ...
    GLOBAL VARIABLES

const float pi = 3.141592653;
float IMU_accel_data_X, IMU_accel_data_Y, IMU_accel_data_Z; // Values ...
    of local IMU accelerometer (m/s^2)
float IMU_gyro_data_X, IMU_gyro_data_Y, IMU_gyro_data_Z;    // Values ...
    of local IMU gyroscope (degrees/s)
float IMU_mag_data_X, IMU_mag_data_Y, IMU_mag_data_Z;       // Values ...
    of local IMU magnetometer (uT)
float Pitch_deg, Roll_deg, Yaw_deg = 0;                     // Pitch, ...
    Roll and Yaw (degrees)
float degree_turn_value;

// Declare variables outside IMU, since variables initiates to 0 ...
    without defining them in function
float Accel_total_vector_modulus;        // Accelerometer total vector ...
    magnitude (m/s^2)
float Accel_pitch_deg, Accel_roll_deg;   // Accelerometer pitch and ...
    roll angles (degrees)
float Gyro_pitch_deg, Gyro_roll_deg;     // Gyroscope pitch and roll ...
    angles (degrees)
bool Gyro_Accel_sync = false;            // Check if Gyroscope and ...
    Accelerometer are synchronized
float Mag_X_calc, Mag_Y_calc;            // Calculated Magnetometer ...
    value (uT)
```

```
float Mag_X_corrected, Mag_Y_corrected; // Corrected value ...
    Magnetometer value (with damping) (uT)
float Pitch_rad, Roll_rad;              // Pitch and Roll angles (rad)

int RW_speed = 0;        // Value of Reaction Wheel speed [from -255 ...
    to 255]
int OBC_mode_value = 0; // Value of On Board Computer mode (waiting, ...
    positioning, detumbling, etc.)
int OBC_data_value = 0; // Value of On Board Computer data (desired ...
    angle turn, etc.)
int Stop_state = 0;      // If Stop_state != 0, motor stops rotating

// PID control definitions
double PID_error, PID_last_error;            // Initialize error and ...
    previousError
double PID_cumulative_error, PID_rate_error; // Initialize the ...
    cumulative Error (Integral) and the rate of Error (Derivative)
float Current_Yaw_deg = 0;                   // Addition of 180 deg ...
    to all values. The error is a substract, so the difference is the ...
    same

double kp = 1.6*1.5; // Proportional contribution
double ki = 0.006*1.5; // Integral contribution
double kd = 0.6*1.5; // Derivative contribution

float Deg_to_reach = 0;  // Setpoint angle (degrees)
bool Zero_state = false; // Check if PID encompass yaw of 0 degrees ...
     (to turn the value to a workable zone)
int PID_output = 0;      // Output value of the PID [from 0 to 255]

// ______________________________________________________________________ ...
    FUNCTIONS
// ______________________________________________________________________ ...
    GENERAL USE FUNCTIONS
// ______________________________________________________________________ ...
    OBC Functions
void OBC_mode_receive()
{
  /*
    Function to set the Bluepill to receiving mode. Uses Timer CH3. ...
        Serial1 as UART communication

    INPUT: None, but gets mode from User Serial1 input
    OUTPUT: None, but saves OBC_mode_value
  */

  // Check if UART Serial1 is available
  read_IMU();

  if (Serial1.available() > 0)
  {
    String bufferString = ""; // String for buffer of Serial1
    // Keep saving input prompt
    while (Serial1.available() > 0)
    {
      bufferString += (char)Serial1.read(); // Adds chars to the ...
         Serial1 buffer
    }
    // Conversion from String to int
    OBC_mode_value = bufferString.toInt();
    Serial1.print("Mode Number: ");
    Serial1.println(OBC_mode_value);
  }
}

void OBC_data_receive()
{
  /*
    Function to set the Bluepill to receiving data mode. Uses Timer ...
       CH3. Serial1 as UART communication

    INPUT: None, but gets data from User Serial1 input
    OUTPUT: Save OBC_data_value
  */
  read_IMU();
  // Check if UART Serial1 is available
  if (Serial1.available() > 0)
  {
    String bufferString = ""; // String for buffer of Serial1
    // Keep saving input prompt
    while (Serial1.available() > 0)
    {
      bufferString += (char)Serial1.read(); // adds chars to the ...
         Serial1 buffer
    }
    OBC_data_value = bufferString.toInt(); // Conversion from String ...
       to int
  }
}

// _________________________________________________ ...
   IMU_Juan_Palomares (commented in Spanish)
void IMU() {
  fifoCount = mpu.getFIFOCount(); //Se obtiene el n mero de bytes ...
     del DMP
  if (fifoCount ≥ 42) { //Si el n mero de datos en el DMP es mayor a ...
     42 bytes (mayor que el packetSize)
    mpu.resetFIFO(); //Se resetea el FIFO
    fifoCount = mpu.getFIFOCount(); //Se vuelven a obtener el n mero ...
       de bytes en el FIFO del DMP
  }
  while (fifoCount < packetSize) { //Mientras el n mero de datos es ...
      menor al paquete
    fifoCount = mpu.getFIFOCount(); //Se vuelven a obtener el n mero ...
        de bytes en el FIFO del DMP
    delay(1); //El delay es necesario pulse_stop controlar el ...
        fifoCount. No se ha podido ver exactamente como afecta porque ...
        al crear un println el problema se resuelve por el incremento ...
        de tiempo.
    if (fifoCount > packetSize) { //Si el n mero de datos en el DMP ...
        es mayor a 42 bytes (mayor que el packetSize)
      mpu.resetFIFO(); //Se resetea el FIFO
      fifoCount = mpu.getFIFOCount(); //Se vuelven a obtener el ...
          n mero de bytes en el FIFO del DMP
    }
  }
  fifoCount = fifoCount - packetSize; //Se resta al fifoCount el ...
      packetSize (el resultado deber a ser siempre 0)
  mpu.getFIFOBytes(fifoBuffer, packetSize); //Se obtienen los valores ...
      del DMP en forma de bytes
  mpu.dmpGetQuaternion(&q, fifoBuffer); //Se obtiene el cuaterni n ...
      de actitud mediante el DMP
  QuatToEuler(q, yaw, pitch, roll); //Se hace el paso del cuaterni n ...
      de actitud a los  ngulos  de Euler

// The range of yaw, pitch adn roll anfles goes from 180 to -180: We ...
    need to adapt it to 0 to 360 range
  if (yaw < 0) yaw += 360;
  if (pitch < 0) pitch += 360;
  if (roll < 0) roll += 360;
// Variable standardization to the ones used in the original program ...
    (the range of these angles goes from 0 to 360)
  Yaw_deg=yaw;
  Pitch_deg=pitch;
  Roll_deg=roll;
}

void QuatToEuler(Quaternion q, float &yaw, float &pitch, float &roll) {
  test = q.x * q.z - q.w * q.y;
  sqx = q.x * q.x;
  sqy = q.y * q.y;
  sqz = q.z * q.z;
  sqw = q.w * q.w;
  roll = (180 / pi) * atan2(2 * q.y * q.z + 2 * q.w * q.x, sqz - sqy ...
      - sqx + sqw);
  pitch = -(180 / pi) * asin(2 * test);
  yaw = (180 / pi) * atan2(2 * q.x * q.y + 2 * q.w * q.z, sqx + sqw - ...
      sqz - sqy);
  //Se asegura que en caso de gimbal lock, el resultado est  definido
  if (pitch ≥ 89.5) {
    pitch = 90;
    yaw = (180 / pi) * 2 * atan2(q.z, q.w);
    roll = 0;
  } else if (pitch ≤ -89.5) {
    pitch = -90;
    yaw = (180 / pi) * 2 * atan2(q.z, q.w);
    roll = 0;
  }
}

// _________________________________________________________________ ...
   Driver Functions
void set_impulse(bool RW_direction, int New_RW_speed)
{
  /*
    Function to set the motor at a fixed direction and speed (B = ...
        port 1, A = port 0)

    INPUT:
      RW_direction: Reaction Wheel direction [Clockwise or Counter ...
          Clockwise] in A
      New_RW_speed: New reaction wheel speed [from 0 to 255]
    OUTPUT:
      None, but saves speed of the Reaction Wheel
  */

  // Account for change in direction
  // If motor's z-axis direction is coincident with CubeSat's z-axis ...
      direction
  /**
   * RW_direction == true  ---> CounterClockwise, motor direction = 0
   * RW_direction == false  ---> Clockwise, motor direction = 1
   *
   * */
  // If motor's z-axis direction is CONTRARY to CubeSat's z-axis ...
      direction
  /**
   * RW_direction == true  ---> CounterClockwise, motor direction = 1
   * RW_direction == false  ---> Clockwise, motor direction = 0
   *
   * */

  if (RW_direction) // Reaction Wheel Counter Clock Wise
  {
    DriverOne.setDrive(1, 0, New_RW_speed); // Change direction ...
        depending on motor connection
    RW_speed = New_RW_speed;
  }
  else // Reaction Wheel Clock Wise
  {
    DriverOne.setDrive(1, 1, New_RW_speed);
    RW_speed = -New_RW_speed;
  }
```

```
}


// ... ...
     IMU Functions
void read_IMU()//The main function of this subroutine is to obtain ...
     the Euler's angles values from the DMP
{
   IMU();
}
void COARSE()
{
   IMU();
     // PROGRAM FOR COARSE POSITIONING
    //Checking if the rotation pass through zero
        if(Yaw_deg_prev-Yaw_deg ≥ 180){
          state_zero_CCW=true;
          }
        else if(Yaw_deg-Yaw_deg_prev ≥ 180){
          state_zero_CW=true;
          }
    //Modification of ∆_degree if state_zero
        if(state_zero_CCW){
          Yaw_deg += 360;
        }
        if(state_zero_CW){
          Yaw_deg -= 360;
        }
    //In case the rotation is CCW or CW ∆_degree calculation is NOT ...
        different
        ∆_degree= abs(Yaw_deg - initial_degree);
    /*    if(not_accel==false && Yaw_deg≥(final_degree-45)){//30 ...
        degree before the tolerance reaching
            Serial1.println("Set zero");
          set_impulse(!CBS_direction, 0);
        } */
        delay(1);
        Yaw_deg_prev=Yaw_deg;
}

void show_IMU()
{
  /*
    Function to show IMU_Data on the Serial1 Monitor

    INPUT:
    None
    OUTPUT:
    None
  */
```

```

//  // Print accelerometer values
////    Serial1.print("Ax: ");
//    Serial1.print(IMU_accel_data_X, 4);
//    Serial1.print(';');
//    Serial1.print(IMU_accel_data_Y, 4);
//    Serial1.print(';');
////   Serial1.print(IMU_accel_data_Z,4);
////   Serial1.println("");
//
//  // Print Gyro values
//  // Serial1.print(" / G: ");
//  // Serial1.print(IMU_gyro_data_X,4);  Serial1.print(';');
//  // Serial1.print(IMU_gyro_data_Y,4);  Serial1.print(';');
//   Serial1.print(IMU_gyro_data_Z,4);   Serial1.print(';');
//  // Serial1.println("");
//
//  // Print Mag values
//  //   Serial1.print("MAGS: ");
//  //   Serial1.print(IMU_mag_data_X,4);    Serial1.print(';');
//  //   Serial1.print(IMU_mag_data_Y,4);    Serial1.print(';');
//  //   Serial1.print(IMU_mag_data_Z,4);
//  //   Serial1.println("");
//
//  // Print orientation angles
////  Serial1.print(" / D: ");
////  Serial1.print(Pitch_deg, 4);
////  Serial1.print(';');
////  Serial1.print(Roll_deg, 4);
////  Serial1.print(';');
//  Serial1.print(Yaw_deg, 4);
//  Serial1.println("");

    Serial1.print(yaw, 4);//Only the Yaw angle is needed for 1DoF ...
        attitude control
    Serial1.println("");
    Serial1.print(';');
//   Serial1.print(pitch, 4);
//   Serial1.println("");
//   Serial1.print(';');
//   Serial1.print(roll, 4);
//   Serial1.println("");

}

void read_show_IMU()
{
  /*
    Function to both read and show IMU_Data.

    INPUT:
```

```
    None
    OUTPUT:
    None, but saves IMU variables and Roll, Pitch and Yaw
  */

  read_IMU();
  show_IMU();
}


// ________________________________________________________________ ...
   PID Functions (Pending tuning)
void computePID()
{
  /*
    Function to calculate PID

    INPUT:
    None
    OUTPUT:
    None, but saves PID values
  */

  // Time step
  float dT = ((1 / ((float)STM32_CLOCK / ...
      (float)Timer1.getPrescaleFactor())) * Timer1.getOverflow()) * ...
      0.001; // in seconds

  read_show_IMU();

  Current_Yaw_deg = Yaw_deg;

  // Check if PID passes through 0 degrees
  if (Zero_state)
  {
    // Saves if the pid pass through 0, if it does it moves the zone ...
       away from 0
    // New variable to not modify the Yaw_deg value
    Current_Yaw_deg = Yaw_deg + 180; // Addition of 180 deg to all ...
       values. The error is a substract, so the difference is the same
  }

  if (Current_Yaw_deg ≥ 360)
    Current_Yaw_deg -= 360; // One of them will increase over 360, it ...
       is a correction

  // Percentage
  volatile float Yaw_deg_perc = (Current_Yaw_deg - 360) / (-360) * 100;

  // Transform angle to percentage (Deg_to_reach == setPoint)
  volatile float Deg_to_reach_perc = (Deg_to_reach - 360) / (-360) * 100;

  // Errors
  PID_error = Deg_to_reach_perc - Yaw_deg_perc;        // Calculate ...
     error (Proportional)
  PID_cumulative_error += PID_error * dT;              // Calculate ...
     the cumulative error (Integral)
  PID_rate_error = (PID_error - PID_last_error) / dT; // Calculate ...
     the rate of error (Derivative)

  // PID Control
  float PID_P = kp * PID_error;             // Proportional
  float PID_I = ki * PID_cumulative_error; // Integral
  float PID_D = kd * PID_rate_error;       // Derivative

  /*
    // Limit
    // Ensure not overflow
    if (PID_P > 255)
      PID_P = 255;
    if (PID_P < -255)
      PID_P = -255;
    if (PID_I > 255)
      PID_I = 255;
    if (PID_I < -255)
      PID_I = -255;
    if (PID_D > 255)
      PID_D = 255;
    if (PID_D < -255)
      PID_D = -255;
  */

  PID_output = PID_P + PID_I + PID_D; //  PID control

  // Prevent overflow
  if (PID_output > 255)
    PID_output = 255;
  if (PID_output < -255)
    PID_output = -255;

  bool RW_direction = true; // true=positive (CounterClockwise), ...
     false=negative (Clockwise)
  Serial1.println(PID_output);

  if (PID_output < 0)
  {
    PID_output = -PID_output;
    RW_direction = false;
  }

  set_impulse(RW_direction, PID_output);

  // Save current error and time for next iteration
  PID_last_error = PID_error; // Save current error
}


// ... ________________________________________________________________________ ...
    MODES OF OPERATION FUNCTIONS
void mode_OBC_Input_Wait()
{
     Serial1.println("Waiting");
  /*
    0. Default mode, OBC Reading

    INPUT:
    None
    OUTPUT:
    None, but exits to mode_Select
  */
  if (Stop_state != 0)
  {
    Stop_state = 0;
    Timer1.detachInterrupt(TIMER_CH3);
  }
  OBC_mode_value = 0;
  Timer1.attachInterrupt(TIMER_CH3, OBC_mode_receive);
  while (OBC_mode_value == 0)
  {
    delay(1); // If not used the while function does not work
    // it can be added more conditions to evade being blocked until a ...
        data is received.
    // for example, it could function an interrupt with a forced exit ...
        and an if after or something
  }
  Timer1.detachInterrupt(TIMER_CH3);
  Serial1.println(OBC_mode_value);
  mode_Select(OBC_mode_value);
}

void mode_Select(int mode_value)
{
  /*
    Function to select Mode

    INPUT:
    mode_value ['0': Waiting; '1': Positioning; '2': Reading]
    OUTPUT:
    None, but exits to selected mode
  */

  switch (mode_value)
  {
  default: // Mode OBC Input Waiting (0): Waiting for OBC
    Serial1.println("Reading mode from OBC");
    mode_OBC_Input_Wait();
    break;
  case 1: // Mode Positioning RW only
    Serial1.println("Mode Positioning");
    mode_Positioning_RW();
    break;
  case 2: // Mode IMU reading (For Testing purposes)
    Serial1.println("Reading IMU");
    mode_IMU_reading();
    break;
  case 3: // Mode to unfold the solar panel
    Serial1.println("Deployment");
    unfolding();
    break;
  case 4: // Mode to fold the solar panel (For Testing purposes)
    Serial1.println("Mode Solar Array");
    folding();
    break;
  }
}
// __________________________________________________________ CASE 1: ...
   COARSE main mode
void mode_Positioning_RW()
{

  OBC_mode_value = 0;
  Serial1.println("Positioning begin");
  set_impulse(CBS_direction, 0);
//Show initial position
  read_IMU();
  Serial1.print("Axis Z position: ");
  Serial1.println(Yaw_deg);

//Set initial speed of the RW
  set_impulse(CBS_direction, initial_speed);
  delay(6000);
  Serial1.println("RW ON INITIAL SPEED");

//Set the desired movement
  Serial1.println("Insert degree value to turn");
  OBC_data_value = 0;
  Timer1.attachInterrupt(TIMER_CH3, OBC_data_receive);
  while (OBC_data_value == 0)
  {
    delay(1); // If not used the while function does not work
  }
  Timer1.detachInterrupt(TIMER_CH3);

```

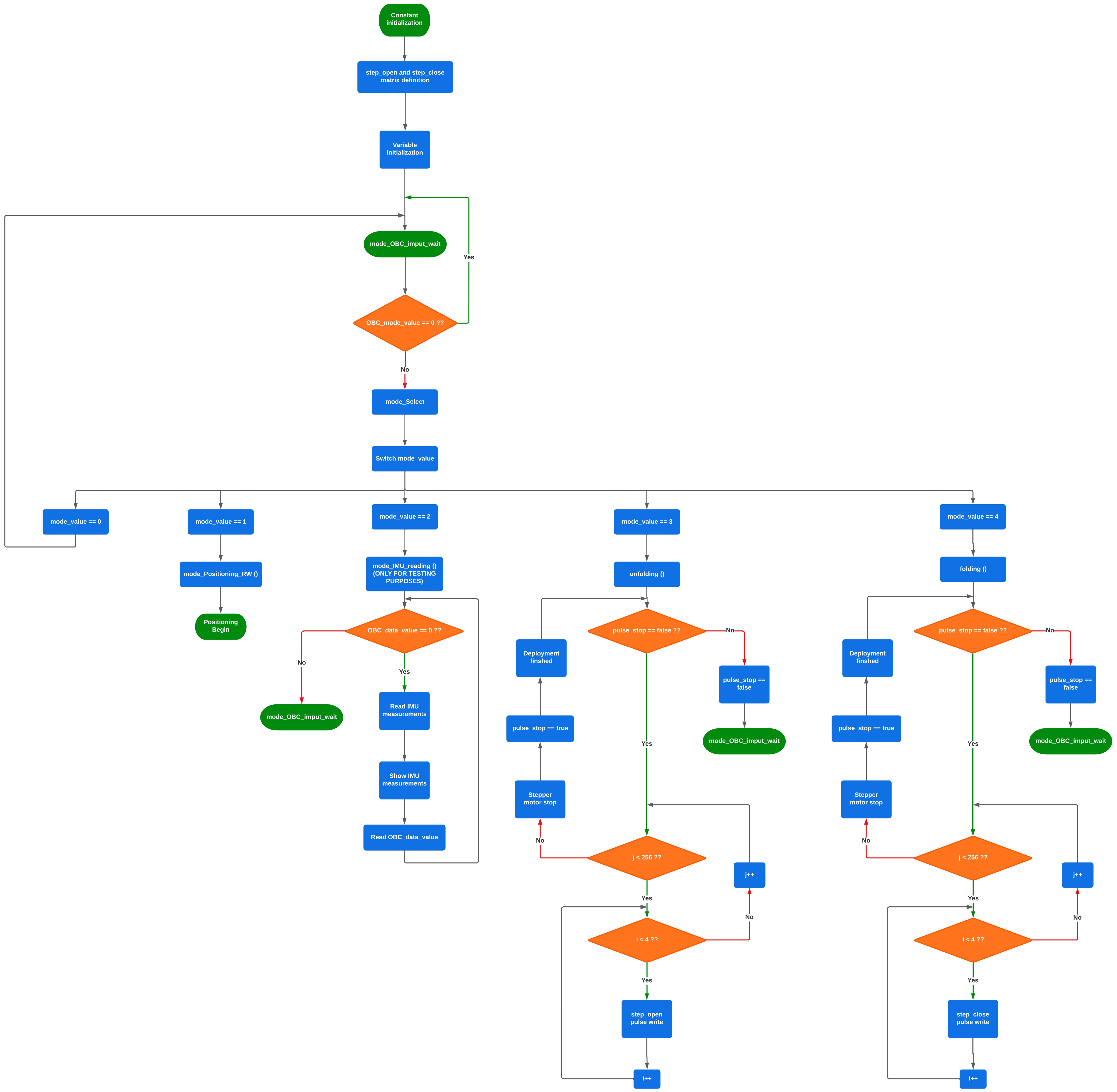
```
//Desired position established
  read_IMU();
  initial_degree= Yaw_deg;
  Yaw_deg_prev=Yaw_deg;
  float degree_turn_value = OBC_data_value;
  Serial1.print("Value to turn: "); // In this case turn, but can be ...
      what is needed
  Serial1.println(OBC_data_value);



//Loop of COARSE positioning________________________________________
  if((△_degree > (abs(degree_turn_value) + Pointing_mode_tolerance)) ...
      ||(△_degree < (abs(degree_turn_value) - Pointing_mode_tolerance))){
//Reinitializaton of CBS_direction
  CBS_direction=false;
//Check if the degree to turn is lower than 0 in order to reverse the ...
    direction of the rotation
  if(degree_turn_value < 0){
    CBS_direction= !CBS_direction;
    }
//Check if the setpoint is bigger than 180  to set the shortest ...
    possible movement
  if(degree_turn_value > 180){
    CBS_direction= !CBS_direction;
    degree_turn_value = 360 - degree_turn_value;
    }
//Accelerates the RW to begin the movement of the device
  Serial1.println("Corse positioning");
  set_impulse(CBS_direction, impulse_speed);
  Serial1.println("Accelerating");


//Evaluates and compare the relative movement in order to stop the ...
    rotation once the maneuver is complete
  Timer1.attachInterrupt(TIMER_CH3, COARSE);
    while(End_Coarse == false)
    {
      for(int u=0;u < 20;u++){
        delay(1);
       }
      if ((abs(degree_turn_value)-△_degree) ≤ Pointing_mode_tolerance)
      {
        End_Coarse=true;
      }
      else End_Coarse=false;
//Prints trough the Bluetooth terminal the relative movement in every ...
    single moment
    //    Serial1.println("Delta");
        Serial1.println(△_degree);
    //    Serial1.println("Yaw");
    //    Serial1.println(Yaw_deg);
    }

  End_Coarse= false;
  state_zero_CW=false;
  state_zero_CCW=false;
//Begin the deceleration of the RW
  set_impulse(!CBS_direction, decelering_speed);
  delay(2000);
//  set_impulse(CBS_direction, initial_speed);
  set_impulse(CBS_direction, final_speed);

  Serial1.println("Deceleraing");
  OBC_data_value = 0;
  for(int k=0;k < 5000;k++){
    Serial1.println(△_degree);
    }
  Timer1.detachInterrupt(TIMER_CH3);
//Show one last value of Yaw angle
  read_IMU();
  Serial1.println("Yaw");
  Serial1.println(Yaw_deg);
  Serial1.println("Delta");
  Serial1.println(△_degree);
  Serial1.println("Coarse_OK");

  △_degree=0;
  mode_OBC_Input_Wait();
}
////FINE (Only when COARSE has ...
    finished)________________________________________________
//  if((△_degree < (degree_turn_value + Pointing_mode_tolerance)) && ...
    (△_degree > (degree_turn_value - Pointing_mode_tolerance))){
//  positioning_Fine();
//  mode_OBC_Input_Wait();
//}
}
// ________________________________________________________ CASE 1: ...
    FINE mode (to improve)
void positioning_Fine()
{
  /*
    1.2. Mode Positioning Fine: PID

    INPUT:
    None
    OUTPUT:
    None, but exits to mode_select
  */

  Serial1.println("Mode Positioning Fine");
```

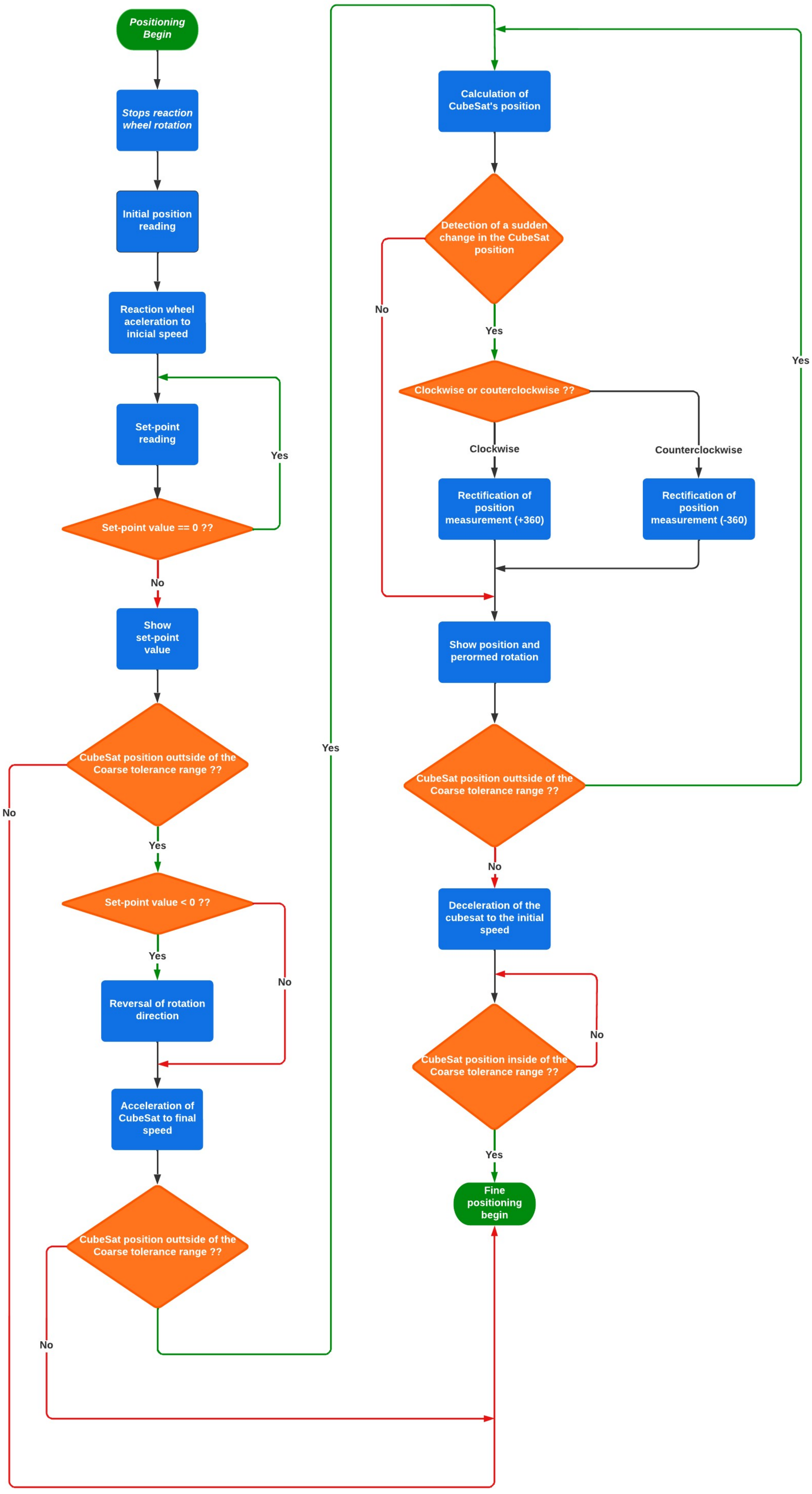
```

bool waiting = true;

Deg_to_reach = OBC_data_value + Yaw_deg; // Get value to reach, ...
    contained in 360
if (Deg_to_reach < 0)
{
  Deg_to_reach += 360;
  Zero_state = true; // Used to store if it passes 0.
  // A PD passing through 0 could give great problems, as it has a ...
      very big change in value. Further used in computePID()
}
else if (Deg_to_reach ≥ 360)
{
  Deg_to_reach -= 360; // These two lines contain the value of the ...
      Yaw_deg in 0-360 degrees.
  Zero_state = true;
}
else
{
  Zero_state = false;
}

if (Zero_state)
{
  Deg_to_reach + 180;
  if (Deg_to_reach ≥ 360)
    Deg_to_reach -= 360;
}

Timer1.attachInterrupt(TIMER_CH4, computePID);
while (waiting)
{ // Range of tolerance
  if (abs(IMU_gyro_data_Z) < Gyro_tolerance && ...
      abs(IMU_accel_data_X) < Accel_tolerance && abs(Deg_to_reach - ...
      Yaw_deg) < Final_pointing_tolerance)
  { // we consider it is stopped, modify values to be accurate
    waiting = false;
  }
  delay(1); // Change
}
waiting = true;
Timer1.detachInterrupt(TIMER_CH4);

// At exit, RW_speed could not be 0
OBC_data_value = 0;
Serial1.println("End of manoeuvre");

// Detach EmergencyStop
Timer1.detachInterrupt(TIMER_CH3);

```

```
}
// ____________________________________________________ CASE 2: Only ...
    MPU reading mode(for testing)
void mode_IMU_reading()
{
  /*
    2- IMU reading (TEST PURPOSE Function)

    INPUT:
    None
    OUTPUT:
    None, but saves IMU variables and Roll, Pitch and Yaw
  */

  OBC_mode_value = 0;
  Timer1.attachInterrupt(TIMER_CH4,read_show_IMU);
  Timer1.attachInterrupt(TIMER_CH3, OBC_data_receive);
  while (OBC_data_value == 0)
  {
    delay(1); // If not used the while function does not work
  }
  Timer1.detachInterrupt(TIMER_CH3);
  Timer1.detachInterrupt(TIMER_CH4);
  OBC_data_value = 0;
  mode_Select(OBC_mode_value);
}
// _____________________________________________________ CASE 3: ...
    Solar panel unfolding mode
void unfolding()
{
 Serial1.println("Unfolding");
 //The assignment of the corresponding pulses to the Stepper motor ...
     inductances begins
 if(pulse_stop==false){
  for(int j = 0; j < 256; j++){
    for (int i = 0; i < 4; i++)
    {
      digitalWrite(IN1, step_open[i][0]);
      digitalWrite(IN2, step_open[i][1]);
      digitalWrite(IN3, step_open[i][2]);
      digitalWrite(IN4, step_open[i][3]);
      delay(speedM);
    }
  }
      digitalWrite(IN1, 0);
      digitalWrite(IN2, 0);
      digitalWrite(IN3, 0);
      digitalWrite(IN4, 0);
      delay(5);
      pulse_stop = true;
      Serial1.println("Deployment finished");
 }
pulse_stop=false;
mode_OBC_Input_Wait();
}
// ______________________________________________________________ CASE ...
    4: Solar panel folding mode
void folding()
{
 Serial1.println("Folding");
 //The assignment of the corresponding pulses to the Stepper motor ...
     inductances begins
 if(pulse_stop==false){
  for(int j = 0; j < 256; j++){
    for (int i = 0; i < 4; i++)
    {
      digitalWrite(IN1, step_close[i][0]);
      digitalWrite(IN2, step_close[i][1]);
      digitalWrite(IN3, step_close[i][2]);
      digitalWrite(IN4, step_close[i][3]);
      delay(speedM);
    }
  }
      digitalWrite(IN1, 0);
      digitalWrite(IN2, 0);
      digitalWrite(IN3, 0);
      digitalWrite(IN4, 0);
      delay(5);
      pulse_stop = true;
      Serial1.println("Deployment finished");
}
pulse_stop=false;
mode_OBC_Input_Wait();
}


// _____________________________________________________________ VOID ...
    SETUP
void setup()
{
  delay(1000); //  wait to let open the serial
// ______________________________________________________________ Timers
  Timer1.pause();
  Timer1.setPrescaleFactor(7200); // 72MHz Clock / 7200 = 10KHz timer
  Timer1.setOverflow(1000);       // Overflow occurs at 1000, each ...
      100 ms timer restarts

  Timer1.setMode(TIMER_CH3, TIMER_OUTPUT_COMPARE); // Configure ...
      channel to OUTPUTCOMPARE: Channel for OBC read values
  Timer1.setMode(TIMER_CH4, TIMER_OUTPUT_COMPARE); // Channel for IMU ...
      read values
  Timer1.setCompare(TIMER_CH3, 1);                   // Phase value in ...
      Overflow range
  Timer1.setCompare(TIMER_CH4, 1);

  Timer1.refresh(); // Refresh timer and start over
  Timer1.resume();

// ______________________________________________________________ ...
   Initiations
  Serial1.begin(9600);
  Wire.begin();

  Serial1.println("START");

// _______________________________________________________________ Setups

  pinMode(LEDPIN, OUTPUT); // Integrated LED
  pinMode(0x68, OUTPUT);   // NCS Pin definition
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

// _______________________________________________________________ ...
   Driver Setup
  Serial1.println("Configuring Driver...");
  DriverOne.settings.commInterface = I2C_MODE; // Driver Comm Mode
  DriverOne.settings.I2CAddress = 0x5D;        // Driver Address ...
      (0x5D by Default)

  while (DriverOne.begin() != 0xA9)
  { // Driver wait for idle
    Serial1.println("ID Mismatch");
    delay(200);
  }
  Serial1.println("ID Match");

  Serial1.println("Waiting for enumeration"); // Driver wait for ...
      peripherals
  while (DriverOne.ready() == false)
    ;
  Serial1.println("Ready");

  while (DriverOne.busy())
    ; // Driver enable
  DriverOne.enable();
  Serial1.println("Driver Ready to Use!");




```

```
// _______________________________________________________ IMU Setup
  //IMU
  mpu.initialize(); //Inicializaci n de la imu
  Serial1.println("MPU successfully inicialized");
  mpu.dmpInitialize(); //DMP initialization
  mpu.setDMPEnabled(true); //DMP is enabled
  Serial1.println("successfull DMP acces");
  packetSize = mpu.dmpGetFIFOPacketSize(); //The DMP paket size is ...
      obtained (42)

  //The offsets calculated through MPU6050_calibration are introduced

  //ORIGINAL CALIBRATION_______________________________________________

//  mpu.setXAccelOffset(-2538);
//  mpu.setYAccelOffset(-1997);
//  mpu.setZAccelOffset(1888);
//  mpu.setXGyroOffset(95);
//  mpu.setYGyroOffset(34);
//  mpu.setZGyroOffset(-45);
//  Serial1.println("MPU calibrated");

  //CALIBRATION TEST 1_________________________________________________

//  mpu.setXAccelOffset(-1045);
//  mpu.setYAccelOffset(-1484);
//  mpu.setZAccelOffset(846);
//  mpu.setXGyroOffset(170);
//  mpu.setYGyroOffset(-64);
//  mpu.setZGyroOffset(36);
//  Serial1.println("MPU calibrated");

  //TEST CALIBRATION 2_________________________________________________

  mpu.setXAccelOffset(-7602);
  mpu.setYAccelOffset(5316);
  mpu.setZAccelOffset(8709);
  mpu.setXGyroOffset(5);
  mpu.setYGyroOffset(-8);
  mpu.setZGyroOffset(-21);
  Serial1.println("MPU calibrated");

  //The initial attitude quaterinon is defined
  q.w = 1;
  q.x = 0;
  q.y = 0;
  q.z = 0;

  delay(100);
  mode_OBC_Input_Wait();

```

```
}

// ___________________________________________________________ ...
   VOID LOOP
void loop()
{
  mode_OBC_Input_Wait();
}
```


Constant initialization
step_open and step_close matrix definition
Variable initialization
mode_OBC_imput_wait
OBC_mode_value == 0 ??
Yes
No
mode_Select
Switch mode_value
mode_value == 0
mode_value == 1
mode_value == 2
mode_value == 3
mode_value == 4
mode_Positioning_RW ()
Positioning Begin
mode_IMU_reading () (ONLY FOR TESTING PURPOSES)
OBC_data_value == 0 ??
No
Yes
mode_OBC_imput_wait
Read IMU measurements
Show IMU measurements
Read OBC_data_value
unfolding ()
Deployment finshed
pulse_stop == false ??
No
pulse_stop == false
mode_OBC_imput_wait
pulse_stop == true
Yes
Stepper motor stop
No
j < 256 ??
j++
Yes
No
i < 4 ??
Yes
step_open pulse write
i++
folding ()
Deployment finshed
pulse_stop == false ??
No
pulse_stop == false
mode_OBC_imput_wait
pulse_stop == true
Yes
Stepper motor stop
No
j < 256 ??
j++
Yes
No
i < 4 ??
Yes
step_close pulse write
i++

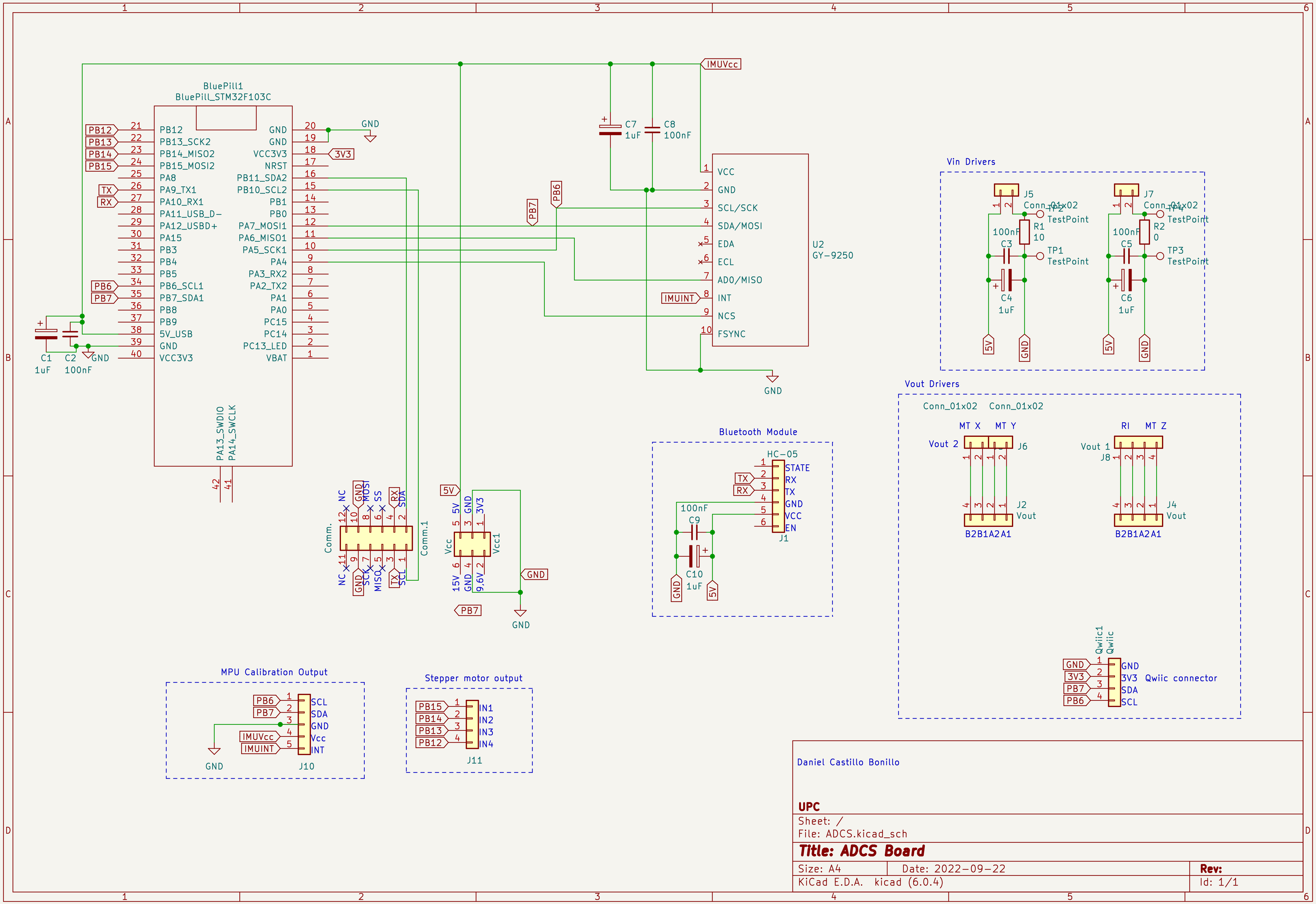

Positioning Begin
Stops reaction wheel rotation
Initial position reading
Reaction wheel aceleration to inicial speed
Set-point reading
Set-point value == 0 ??
Yes
No
Show set-point value
CubeSat position outtside of the Coarse tolerance range ??
No
Yes
Set-point value < 0 ??
Yes
No
Reversal of rotation direction
Acceleration of CubeSat to final speed
CubeSat position outtside of the Coarse tolerance range ??
No
Yes
Calculation of CubeSat's position
Detection of a sudden change in the CubeSat position
No
Yes
Clockwise or couterclockwise ??
Clockwise
Counterclockwise
Rectification of position measurement (+360)
Rectification of position measurement (-360)
Show position and peromed rotation
CubeSat position outtside of the Coarse tolerance range ??
Yes
No
Deceleration of the cubesat to the initial speed
CubeSat position inside of the Coarse tolerance range ??
No
Yes
Fine positioning begin

D.2 MPU calibration code

```
// Arduino sketch that returns calibration offsets for MPU6050
//   Version 1.1  (31th January 2014)
// Done by Luis R denas <luisrodenaslorda@gmail.com>
// Based on the I2Cdev library and previous work by Jeff Rowberg ...
    <jeff@rowberg.net>
// Updates (of the library) should (hopefully) always be available at
// https://github.com/jrowberg/i2cdevlib

// These offsets were meant to calibrate MPU6050's internal DMP, but ...
    can be also useful for reading sensors.
// The effect of temperature has not been taken into account so I ...
    can't promise that it will work if you calibrate indoors and then ...
    use it outdoors. Best is to calibrate and use at the same room ...
    temperature.

/* ==========  LICENSE  ==================================
 I2Cdev device library code is placed under the MIT license
 Copyright (c) 2011 Jeff Rowberg

 Permission is hereby granted, free of charge, to any person ...
     obtaining a copy
 of this software and associated documentation files (the ...
     "Software"), to deal
 in the Software without restriction, including without limitation ...
     the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or ...
     sell
 copies of the Software, and to permit persons to whom the Software is
 furnished to do so, subject to the following conditions:

 The above copyright notice and this permission notice shall be ...
     included in
 all copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, ...
     EXPRESS OR
 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT ...
     SHALL THE
 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ...
     ARISING FROM,
 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER ...
     DEALINGS IN
 THE SOFTWARE.
 =========================================================
 */

// I2Cdev and MPU6050 must be installed as libraries
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

///////////////////////////////////   CONFIGURATION   ...
    /////////////////////////////
//Change this 3 variables if you want to fine tune the skecth to your ...
    needs.
int buffersize=1000;     //Amount of readings used to average, make ...
    it higher to get more precision but sketch will be slower   ...
    (default:1000)
int acel_deadzone=8;     //Acelerometer error allowed, make it lower ...
    to get more precision, but sketch may not converge  (default:8)
int giro_deadzone=1;     //Giro error allowed, make it lower to get ...
    more precision, but sketch may not converge  (default:1)

// default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
//MPU6050 accelgyro;
MPU6050 accelgyro(0x68); // <-- use for AD0 high

int16_t ax, ay, az,gx, gy, gz;

int mean_ax,mean_ay,mean_az,mean_gx,mean_gy,mean_gz,state=0;
int ax_offset,ay_offset,az_offset,gx_offset,gy_offset,gz_offset;

///////////////////////////////////   SETUP   ...
    ////////////////////////////////////
void setup() {
  // join I2C bus (I2Cdev library doesn't do this automatically)
  Wire.begin();
  // COMMENT NEXT LINE IF YOU ARE USING ARDUINO DUE
  TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz). Leonardo ...
     measured 250kHz.

  // initialize serial communication
  Serial.begin(115200);

  // initialize device
  accelgyro.initialize();

  // wait for ready
  while (Serial.available() && Serial.read()); // empty buffer
  while (!Serial.available()){
    Serial.println(F("Send any character to start sketch.\n"));
    delay(1500);
  }
  while (Serial.available() && Serial.read()); // empty buffer again

  // start message
  Serial.println("\nMPU6050 Calibration Sketch");
  delay(2000);
  Serial.println("\nYour MPU6050 should be placed in horizontal ...
     position, with package letters facing up. \nDon't touch it until ...
     you see a finish message.\n");
  delay(3000);
  // verify connection
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection ...
     successful" :
  "MPU6050 connection failed");
  delay(1000);
  // reset offsets
  accelgyro.setXAccelOffset(0);
  accelgyro.setYAccelOffset(0);
  accelgyro.setZAccelOffset(0);
  accelgyro.setXGyroOffset(0);
  accelgyro.setYGyroOffset(0);
  accelgyro.setZGyroOffset(0);
}

///////////////////////////////////   LOOP   ...
   ////////////////////////////////////
void loop() {
  if (state==0){
    Serial.println("\nReading sensors for first time...");
    meansensors();
    state++;
    delay(1000);
  }

  if (state==1) {
    Serial.println("\nCalculating offsets...");
    calibration();
    state++;
    delay(1000);
  }

  if (state==2) {
    meansensors();
    Serial.println("\nFINISHED!");
    Serial.print("\nSensor readings with offsets:\t");
    Serial.print(mean_ax);
    Serial.print("\t");
    Serial.print(mean_ay);
    Serial.print("\t");
    Serial.print(mean_az);
    Serial.print("\t");
    Serial.print(mean_gx);
    Serial.print("\t");
    Serial.print(mean_gy);
    Serial.print("\t");
    Serial.println(mean_gz);
    Serial.print("Your offsets:\t");
    Serial.print(ax_offset);
    Serial.print("\t");
    Serial.print(ay_offset);
    Serial.print("\t");
    Serial.print(az_offset);
    Serial.print("\t");
    Serial.print(gx_offset);
    Serial.print("\t");
    Serial.print(gy_offset);
    Serial.print("\t");
    Serial.println(gz_offset);
    Serial.println("\nData is printed as: acelX acelY acelZ giroX ...
       giroY giroZ");
    Serial.println("Check that your sensor readings are close to 0 0 ...
       16384 0 0 0");
    Serial.println("If calibration was succesful write down your ...
       offsets so you can set them in your projects using something ...
       similar to mpu.setXAccelOffset(youroffset)");
    while (1);
  }
}

/////////////////////////////////////   FUNCTIONS   ...
    ////////////////////////////////////
void meansensors(){
  long i=0,buff_ax=0,buff_ay=0,buff_az=0,buff_gx=0,buff_gy=0,buff_gz=0;

  while (i<(buffersize+101)){
    // read raw accel/gyro measurements from device
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

    if (i>100 && i≤(buffersize+100)){ //First 100 measures are discarded
      buff_ax=buff_ax+ax;
      buff_ay=buff_ay+ay;
      buff_az=buff_az+az;
      buff_gx=buff_gx+gx;
      buff_gy=buff_gy+gy;
      buff_gz=buff_gz+gz;
    }
    if (i==(buffersize+100)){
      mean_ax=buff_ax/buffersize;
      mean_ay=buff_ay/buffersize;
      mean_az=buff_az/buffersize;
      mean_gx=buff_gx/buffersize;
      mean_gy=buff_gy/buffersize;
      mean_gz=buff_gz/buffersize;
    }
    i++;
    delay(2); //Needed so we don't get repeated measures
  }
}

void calibration(){
  ax_offset=-mean_ax/8;
  ay_offset=-mean_ay/8;
  az_offset=(16384-mean_az)/8;

  gx_offset=-mean_gx/4;
  gy_offset=-mean_gy/4;
  gz_offset=-mean_gz/4;
  while (1){
    int ready=0;
    accelgyro.setXAccelOffset(ax_offset);
    accelgyro.setYAccelOffset(ay_offset);
    accelgyro.setZAccelOffset(az_offset);

    accelgyro.setXGyroOffset(gx_offset);
    accelgyro.setYGyroOffset(gy_offset);
    accelgyro.setZGyroOffset(gz_offset);

    meansensors();
    Serial.println("...");

    if (abs(mean_ax)≤acel_deadzone) ready++;
    else ax_offset=ax_offset-mean_ax/acel_deadzone;

    if (abs(mean_ay)≤acel_deadzone) ready++;
    else ay_offset=ay_offset-mean_ay/acel_deadzone;

    if (abs(16384-mean_az)≤acel_deadzone) ready++;
    else az_offset=az_offset+(16384-mean_az)/acel_deadzone;

    if (abs(mean_gx)≤giro_deadzone) ready++;
    else gx_offset=gx_offset-mean_gx/(giro_deadzone+1);

    if (abs(mean_gy)≤giro_deadzone) ready++;
    else gy_offset=gy_offset-mean_gy/(giro_deadzone+1);

    if (abs(mean_gz)≤giro_deadzone) ready++;
    else gz_offset=gz_offset-mean_gz/(giro_deadzone+1);

    if (ready==6) break;
  }
}
```

Appendix E

PCB electronic schemes



Daniel Castillo Bonillo

UPC

Sheet: /

File: ADCS.kicad_sch

Title: ADCS Board

Size: A4   Date: 2022−09−22   **Rev:**

KiCad E.D.A. kicad (6.0.4)   Id: 1/1

Appendix F

Datasheet of used components

F.1 MPU-9250