Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA

UPC ^{3}CAT-1

Design and integration of a nanosatellite

October 2012 – October 2013

Author:
Sasha Surroca Tarradas

Director:
Eduard Alarcón Cot

Codirectors:
Roger Jové Casulleras

Abstract

This document is a report of the final project conducted between the months of October 2012 and July 2013 on the integration, both physical and electrical of a CubeSat, in this case of the ^{3}CAT, a project developed by the Electronics Department (EEL) and the Signal Theory and Communications department (TSC) of ETSETB school in the UPC.

The project described needed a long learning process about the Cubesat's standard and about every subsystem that composes the satellite, many of them already being designed when this project started.

The pages that follow the timeline with which this project was carried out. The first pages give a brief introduction to the CubeSat standard and the various subsystems consisting of the ^{3}CAT, followed by the analysis of the environmental conditions that should be tolerated by the satellite and its physical features. The last section explains how, from the analysis mentioned above, decisions have been taken and satellite parts have been designed to achieve compatibility between the subsystems and the required system's functionality.

Although at the time of completion of this project, the ^{3}CAT is not finished, the results show that the mission is perfectly feasible and is on track to finish the integration of the entire system on time to test it and have it ready for the launch date, scheduled for spring 2014.

Abstract

Aquest document és un informe del projecte final de carrera realitzat entre els mesos d'Octubre de 2012 i Juliol de 2013 sobre la integració, tant a nivell físic com elèctric i de dades d'un Cubesat, en aquest cas el ^{3}CAT, un projecte que forma part del Departament d'Electrònica (EEL) i de Teoria de Senyal i Comunicacions (TSC) de l'escola ETSETB de la UPC.

El projecte descrit ha necessitat d'un llarg procés d'aprenentatge en quan a l'estàndard dels Cubesats i en quan als diferents subsistemes que composen el satèl·lit, molts d'ells ja començats a dissenyar quan es va iniciar aquest projecte.

Les pàgines que vénen a continuació segueixen l'estructura del timeline real amb el que s'ha portat a terme aquest projecte. Les primeres pàgines dónen una breu introducció a l'estàndard del Cubesat i als diferents subsistemes dels que consta el ^{3}CAT, seguides de l'anàlisi i l'estudi de les condicions ambientals amb les que es trobarà el satèl·lit així com les seves caràcterístiques físiques. El darrer apartat del treball explica com a partir de l'anàlisi esmentat anteriorment, s'han près decisions i s'han dissenyat algunes parts del satèl·lit per aconseguir la compatibilitat entre tots els subsistemes i també el bon funcionament del sistema complet.

Tot i que en el moment de finalització d'aquest projecte, el ^{3}CAT no està acabat, els resultats obtinguts mostren que la missió és perfectament realitzable i que es va per bon camí per poder acabar la integració real de tot el sistema a temps per poder fer els tests corresponents i tenir-lo preparat per la data del llançament, programat per la primavera de 2014.

Abstract

Este documento es un informe del proyecto final de carrera realizado entre los meses de Octubre de 2012 y Julio de 2013 sobre la integración, tanto a nivel físico como eléctrico y de datos de un CubeSat, en este caso el ^{3}CAT, un proyecto que forma parte del Departamento de Electrónica (EEL) y de Teoría de la Señal y Comunicaciones (TSC) de la escuela ETSETB de la UPC.

El proyecto descrito hay requerido de un largo proceso de aprendizaje en cuanto a los estándards de los Cubesats y en cuanto a los diferentes subsistemas que componen el satélite, muchos de ellos ya empezados a diseñar cuando se inició este proyecto.

Las páginas que vienen a continuación siguen la estructura del timeline real con el que se ha llevado a cabo este proyecto. Las primeras páginas dan una breve introducción al estándar del CubeSat y a los diferentes subsistemas de los que consta el ^{3}CAT, seguidas del análisis y el estudio de las condiciones ambientales con las que se encontrará el satélite así como sus características físicas. El último apartado del trabajo explica cómo a partir del análisis anterior, se han tomado decisiones y se han diseñado algunas partes del satélite para lograr la compatibilidad entre todos los subsistemas y también el buen funcionamiento del sistema completo.

Aunque en el momento de finalización de este proyecto, el ^{3}CAT no está terminado, los resultados obtenidos muestran que la misión es perfectamente realizable y que se va por buen camino para poder terminar la integración real de todo el sistema a tiempo para poder hacer los tests correspondientes y tenerlo preparado para la fecha del lanzamiento, programado para la primavera de 2014.

Acknowledgements

I would like to thank, first of all, Professor Eduard Alarcón for giving me this huge opportunity, not only for accepting me in the ^{3}CAT team but also for betting on me as the system integrator, which is a hard work and usually only suitable for experts.

First, I want to thank the codirectors of this project, Elisenda Bou and Roger Jove as they have always been there to help me when I was not able to solve some problems and for sharing their knowledge about matters that were out of my range. To Elisenda, thank you for encouraging me to take decisions and giving me the trustfulness to take them without hesitating whether they were right or not. And to Roger, thank you for showing me the way to follow at the beginning of the project and for working with me on the thermal analysis, which has taken us a lot of time.

I also want to thank all the ^{3}CAT team (Carles Aragúz, Joan Muñoz, David Vidal, Marc Marí, Sumit Karki, Hernan and Alberto Saez) as they have been available when needed to have a meeting and that have always brought me the necessary information of their systems to be used to design the whole satellite.

I would not like to finish without saying anything about the PAE students that have been working with us for the last 6 months and have a special mention to Mario Gómez, whom I worked together to develop the mission analysis using the STK software.

CONTENTS

CHAPTER 1: INTRODUCTION

CubeCat-1 is the first project of the Universitat Politècnica de Catalunya to develop a pico-satellite. Since the beginnings, this project has been defined as an educational and technological demonstrator planned to be developed by students.

Its primary objective is educational. It is involved in the Conceive Design Implement and Operate (CDIO) study plan of the Telecom Barcelona [1]. This plan started on 2008 and was awarded as the best collective teaching initiative by the Generalitat de Catalunya on 2012 [2]. Apart from the educational component, CubeCat-1 project has been thought as a platform to perform and demonstrate technologies in space. Eleven different experiments and demonstrators are boarded on it. They are listed and briefly explained in sections coming.

1.1 HISTORY

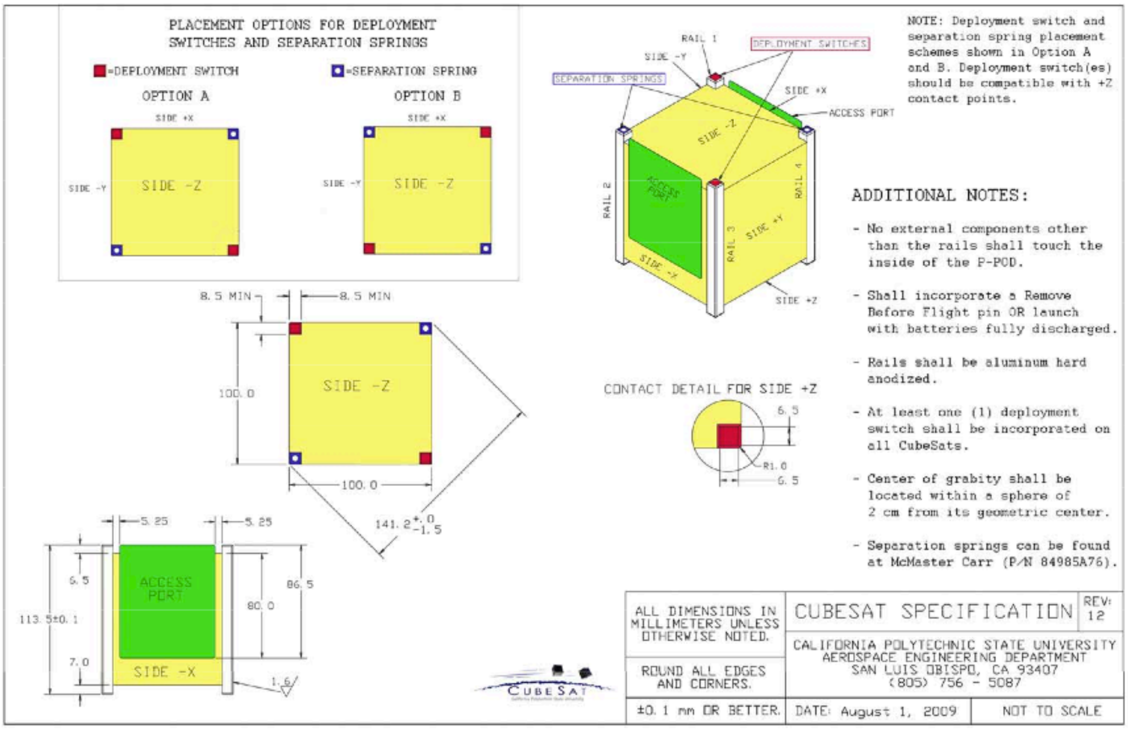
Started in 1999, the CubeSat Project[3] began as a collaborative effort between Prof. Jordi Puig-Suari at California Polytechnic State University (Cal Poly), San Luis Obispo, and Prof. Bob Twiggs at Stanford University's Space Systems Development Laboratory (SSDL). The purpose of the project is to provide a standard for design of pico satellites to reduce cost and development time, increase accessibility to space, and sustain frequent launches. Presently, the CubeSat Project is an international collaboration of over 100 universities, high schools, and private firms developing pico-satellites containing scientific, private, and government payloads. A CubeSat is a 10 cm cube with a mass of up to 1.33 kg. Developers benefit from the sharing of information within the community. [3]

The term "CubeSat" was coined to denote picosatellites that adhere to the standards described in the CubeSat design specification. Cal Poly published the standard in an effort led by Puig-Suari in 1999 while Twiggs has contributed to the CubeSat community, focusing his efforts on CubeSats from educational institutions. In 2004, with their relatively small size, CubeSats could each be made and launched for an estimated $80,000 - $120,000. This price tag, far lower than most satellite missions, has made CubeSat a viable option for schools and universities across the world. Because of this, a large number of universities, some companies and government organizations around the world are developing CubeSats -between 40 and 50 universities were developing CubeSats in 2004, Cal Poly reported.

1.2 CUBESAT STANDARD

At the time CubeSat was developed, picosatellites needed a standard for two reasons: make accessible to space groups or universities that had no experience on it, and lighten the costs and development times of bringing a mission into space. Both would eventually contribute to make space missions more reliable and frequent, and make a huge step forward in this field by opening the door to the universities around the globe to make scientific experiments at a relatively lower cost.

The standard was made as simple as possible, because the aim was to focus on the mission itself, not on the implementation of the standard. Moreover, having a simple standard allows developers to make its own differentiation, having imposed only structural -regarding measures and mass distribution-, electrical and operational restrictions -mainly derived from the fitting of many CubeSats into a launch vehicle. This standard also allowed some companies to be introduced in this new market, like Pumpkin, Isis or Clyde Space.



NOTE: Deployment switch and separation spring placement schemes shown in Option A and B. Deployment switch(es) should be compatible with +Z contact points.

ADDITIONAL NOTES:

- No external components other than the rails shall touch the inside of the P-POD.
- Shall incorporate a Remove Before Flight pin OR launch with batteries fully discharged.
- Rails shall be aluminum hard anodized.
- At least one (1) deployment switch shall be incorporated on all CubeSats.
- Center of grabity shall be located within a sphere of 2 cm from its geometric center.
- Separation springs can be found at McMaster Carr (P/N 84985A76).

ALL DIMENSIONS IN MILLIMETERS UNLESS OTHERWISE NOTED.	CUBESAT SPECIFICATION	REV: 12
ROUND ALL EDGES AND CORNERS.	CALIFORNIA POLYTECHNIC STATE UNIVERSITY AEROSPACE ENGINEERING DEPARTMENT SAN LUIS OBISPO, CA 93407 (805) 756 - 5087	
±0.1 mm OR BETTER.	DATE: August 1, 2009	NOT TO SCALE

Figure 1.1 Cubesat design specifications [3]

1.2.1 Cubesat requirements

[3]**General Requirements**

- All parts shall remain attached to the CubeSats during launch, ejection and operation. No additional space debris shall be created
- No pyrotechnics are permitted
- Total stored chemical energy shall not exceed 100 W·h

Mechanical Requirements

- The CubeSat shall be 100:0 _ 0:1**mm** wide (X and Y) and 113:5 _ 0:1**mm** tall (Z).
- All components shall not exceed 6:5**mm** normal to the CubeSat surface.
- Deployables shall be constrained by the CubeSat. The P-POD rails and walls shall not be used to constrain deployables.
- Each CubeSat shall not exceed 1.33 kg mass.
- The centre of gravity shall be located within a sphere of 2 cm from its geometric centre.

Electrical Requirements

- No electronics shall be active during launch to prevent any electrical or RF interference with the launch vehicle and primary payloads. CubeSats with batteries shall be fully deactivated during launch (when inside the PicoSatellite Orbital Deployer (POD)) or be launched with discharged batteries.
- The CubeSat shall include at least one Deployment Switch (DS) -two for ESA Flights to completely turn off satellite power once actuated. All systems shall be turned off, including real-time clocks.
- The CubeSat shall include a Remove-Before-Flight (RBF) switch or launch with batteries fully discharged. The RBF pin shall be removed from the CubeSat after integration into the P-POD. The RBF pin shall cut all power to the satellite once it is inserted into the satellite, and shall not protrude more than 6.5 mm from the rails when it is fully inserted into the satellite.

Operational Requirements

- CubeSats with batteries shall have the capability to receive a transmitter shutdown command.
- All deployable such as booms, antennas and solar panels shall wait to deploy a minimum of 30 minutes after the CubeSat's Deployment Switch(es) are

activated from the P-POD ejection.

- RF transmitters greater than 1 mW shall wait a minimum of 30 minutes after the Cubesat's deployment switch(es) are activated from P-POD ejection.
- The orbital decay shall be less than 25 years after end of mission life.

Bibliography (Chapter 1)

[1] H Heidt, J Puig-Suari, AS Moore, S Nakasuka, R J Twiggs, CubeSat: A new Generation of Picosatellite for Education and Industry Low Cost Space Experimentation. **Proceedings of the** 14th**Annual/USU Conference on Small Satellites, 2001.**

[2] R Bragos, E Alarcon, A Camps, J Pegueroles, J Sarda, E Sayrol, Design of the Advanced Engineering Project course for the third year of electrical engineering at Telecom BCN **Proceedings of the** 8th **International CDIO Conference, Queensland University of Technology, Brisbane, July 1 - 4, 2012**

[3] "Cubesat Design Specification", Tech. Rep., California Polytechnic State University, 2009, URL: www.cubesat.org

[4] Schilling, K., "Design of pico-satellites for education in systems engineering," *Aerospace and Electronic Systems Magazine, IEEE* , vol.21, no.7, pp.S_9,S_14, July 2006

[5] **A. Camps,** PAE PICOSATS presentation (Chapter 1)

[6] http://www.spacelaunchreport.com/reliability2009.txtç

[7] http://www.aero.org/publications/crosslink/winter2001/images/03_06.gif

[8] **P Fortescue, J Stark, G Swinerd,** Spacecraft Systems Engineering, 2003, third edition

CHAPTER 2: SYSTEM DESCRIPTION

CubeCat-1 is a 1U CubeSat, so it complies with CubeSat Design Specification [1]. It is developed as an educational project in the UPC University. This project involves people who go from Professors to first year students, having graduates, post-graduates and people writing their degree or master thesis. When one wants a cubesat to develop and try their payloads, space-dedicated companies can design it, but its cost is very expensive. What ^{3}CAT is about is to develop the full system for a satellite but without using space-qualified components, it is designed using normal electronic and mechanichal components.

^{3}CAT-1 has many different subsystems and payloads. The difference between them relapses in that the subsystems are those parts that are assential for operation of the cubesat and that every satellite has, and the payloads are those experiments that need to be tested in space environment and that can be different in every mission. The next lines explain both the subsystems and the payloads in the ^{3}CAT, that although some of them work like mamny others, they have been designed for this specific satellite to cover all needs.

2.1 SUBSYSTEMS

The subsystems are those devices that are imprescindible in the satellite for its operation. These subsystems are those that distribute the energy around the satellite, that send and receive information to and from Earth and the main computer.

2.1.1 OBC and software

The On-Board Computer is the board PortuxG20 [2] from Taskit. It has never been tested in space before but it has industrial qualification and has already been tested for a week in the Thermal Vacuum Chamber (TVAC) at a pressure of 10−8 bar. This board offers a good compromise between performance and power consumption having different modes to save energy if the satellite needs it. It was chosen as it has the possibility to run Linux OS on top and for its performance as it can use different energy modes, including those designed by software (disabling interfaces, reducing freqüency…). Furthermore, it has many interfaces available as seen in the list below:

- 10/100 Ethernet MAC
- USB 2.0 Full Speed (12 Mbit/s)
- 2 x USB host
- 1 x USB device
- 1 serial interface (USART / UART)

- Micro SD-Card slot
- JTAG
- DBGU
- 1 x SSC
- 1 x SPI
- TWI (I2C compatible)
- Up to 64 digital I/O ports
- 16-bit Parallel Bus
- Up to 5 serial Interfaces (USART / UART)
- External SD-Card Interface
- 4-channel 10-bit ADC

With an ARM9 400MHz CPU core and a 8Gb external SD card, it is able to run Unix/Debian as operating system making all the resources of OpenSource community available. Finally, it is also possible to disable the interfaces by software in order to save as much energy as possible.

Figure 2.1 Portux G20

From a system level perspective, the CubeCat-1 software architecture has to be defined following a certain functionality guidelines and allowing for the hardware and software constraints. However, it is not only the satellite's functionality or capabilities that have to be taken into account, but also robustness and reliability.

The satellite software has to perform two main tasks: system control and payloads management. While the system control task has to do with energy management, communication subsystem handling and system state control; the payloads management task will be responsible of scheduling the corresponding modules and manage payload's processes and their data (i.e. receive, send, store, compress). This software architecture was chosen to be modular allong with the design objectives of the ^{3}CAT. In a complex system such as this, each functional unit consumes computation time. Since the computation effort is directly related with the power consumption, it clearly has to be minimized. Sometimes, this computation effort generates a specific data that has to be used by other functional units. Therefore, a relationship between the modules is created. One may call this relationship "intermodule-collaboration". In such a scenario, the software has to provide some kind of mechanisms (e.g. data sharing spaces) in order to reduce the computation effort done when twice more than one functional unit needs this data. Since the software will run on a standalone machine, the energy consumption, which is of course critical in a space application like this, has to be taken into account at a system level too. Although the system does handle each module reading the energy level and estimating their consumption, and will not let a module to run if, for instance, there is not enough energy reservoir available, the system itself consumes power and has to be able to self-manage its own power demands.

In a space environment the ability of a system to recover from unexpected states, to correct errors and to transmit the current state is critical. The software has to provide enough mechanisms to let the system detect and correct a certain malfunctioning, and to return to a safe state (and remain there if needed). As a general rule of thumb, the simpler the system, the less likely an error occurrence will be. Even though, the software core has to be designed in such a way in which data loss or spurious states are very unlikely to happen. Apart from watchdogs to reset the system in case it hangs, an external reset will power down the On-board computer at least once a week.

2.1.2 EPS

The Electrical Power Subsystem (EPS) is the module that powers all the satellite subsystems. For this reason, it must be as robust and reliable as possible. It is a critical subsystem since a failure of the EPS means the failure of the whole mission. Furthermore, EPS must be as efficient as possible as the available energy is totally limited. After having had some problems with commercial Pumpkin and Clyde Space Electrical Power boards, the CubeCat-1 team decided to make its own design. The EPS architecture is shown in figure 5 and deeply explained in the next lines.

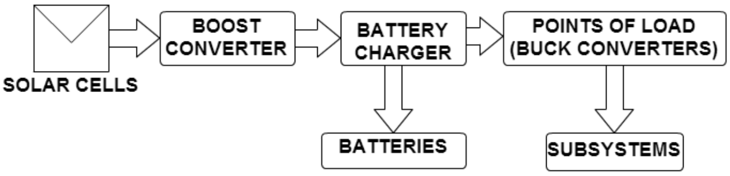


Figure 2.2 EPS architecture

The energy is going to be taken from the solar cells attached on the six cube faces. Usually the solar cells are from the same technology and have the same characteristics. But this is not the case. Five of the six faces are SPECTROLAB TASC solar cells but the remaining one is a CELSAT designed and built by the Electronic Department of UPC. This last array of solar cells apart from being used as a power source is one of the payloads of the mission. The interconnection of all solar cells is in three parallel arrays of two solar cells in series. Each array is composed of two panels from opposite faces of the cube. This is because when one face is illuminated the opposite is not and vice versa. Bypassing with one parallel diode to each solar cell, the energy produced in one panel is not consumed in the opposite one. Moreover, after each array of solar cells a boost converter (commercial product TPS611702) is connected in order to adapt the voltage of each array to the expected one of the rest of the EPS circuit. All this architecture is represented in figure 2.3.

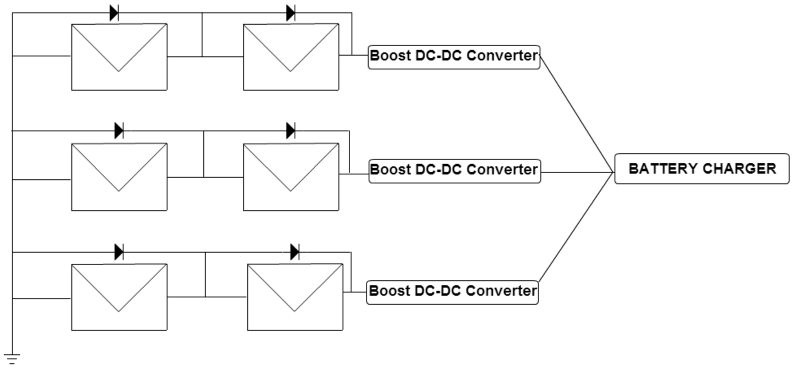


Figure 2.3 Solar cells architecture

The most important goal was making up architecture of heterogeneous solar cells. The CELSAT cells have 7.117 V of open circuit voltage but the SPECTROLAB ones have 10.08 V. For this reason, it is totally necessary to adapt the electrical power to a value of tension that fits both the battery charger and the Points of Load (POL). The mentioned boost DC-DC converter accomplishes this. It adapts the tension coming from the solar cells that ranges between 0 and 10 V to 8.5 V that fits both the BC and the POL. The PCB of the POL has to be as close as possible to the load; for this reason, it is going to be in each subsystem floor. The other part of the EPS system must fit in just one floor: Boost converters, battery charger, batteries and heaters.

The battery charger has to charge up batteries when the solar panels are illuminated. The circuit is based on the MAX17573. The charging process is not always the same, it has different states: at first it charges batteries with a higher voltage than the nominal one and, finally, with this voltage it charges at constant current. This process has to be followed as it increments the batteries expected life and decrements the time needed to charge them. The batteries are an array of two series lithium batteries (DRF604) that have got 7.4 V of nominal tension (3.7 V each one) and a nominal power of 8.51 W (4.255 W each one). Obviously, the available output tension from the batteries is not always the same; it will start from 7.4 V and may slowly go down. The fact is that the charging state of the batteries has to be carefully analysed as it indicates the available power and, what is more, the payloads that can be activated and the ones that cannot. The energy state system can be in three different modes. Firstly, solar cells can be feeding the POL and charging the batteries at the same time. Secondly, batteries can be feeding all POL, as the solar panels are shadowed. And finally, if the power coming from the solar cells is not enough for feeding the POL when they are illuminated, batteries will help giving the needed power to cover the deficit. Finally, the last step is to adapt the voltage coming from the batteries or the solar panels to the needed one of each satellite subsystem. For example, some systems may need 3, 3.3 or 5 V instead of the 7.4 V of the batteries or the 8.5 V of the solar cells array. This is done with a buck DC-DC converter that simply changing its feedback resistors can adapt the input tension to the expected output one (always lower). The commercial buck converter selected to do this conversion is the LTC36045.

A very important consideration is the temperature the satellite can reach. At very low temperatures (bellow 0°C) the batteries can freeze resulting into an irreversible damage to them. So some heaters have to be installed in order to turn up the batteries temperature when they are too cold. The commercial heaters selected to accomplish this duty are the OMEGALUXKHLV6. An NTC resistor connected with a series resistor will detect the temperature. Its voltage will be compared using a comparator with a fixed tension value. When the temperature is too cold, the output of the comparator will enable a driver of current that will feed the heaters situated between the two batteries.

2.1.3 Communications

The telemetry, tracking and communications (TT&C) subsystem consists of a main transceiver to establish the communication link with the ground station and the satellite, a beacon transmitter to transmit the CubeSat identifier and the state of charge of the batteries, and a main power independent Peltier fed beacon to transmit the status of the Peltier cells. The overall Communications scheme is shown in figure 2.4.

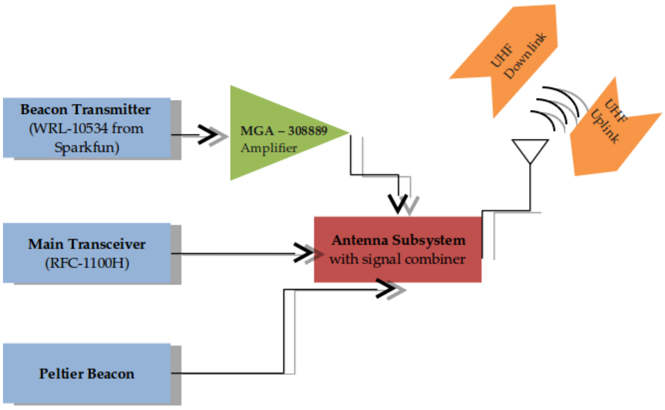


Figure 2.4 Block diagram of the communication system

Main transceiver

A transceiver unit along with all the necessary RF components to output a power of 33dBm does the function of carrying out the main communication link with the ground stations. RFC1100H has been used as the main transceiver unit. It consists of a CC1100 transceiver chip from Texas instruments, and it is responsible for encapsulating the data received from the OBC with necessary error correction. The data packets are then modulated and amplified to deliver a peak power output of 33 dBm. The On-Board computer, Portux G-20, controls this transceiver and communicates with the transceiver unit using SPI. The OBC also implements the transceiver’s finite state machine in order to carry out its basic operations like transmit and receive as well as switching to the different power states such as idle or sleep modes. The payload data is also sent by the OBC to the transceiver.

Beacon signal

The beacon transmitter transmits the CubeSat identifier and the state of the power sources like the one of the batteries and of the solar cells. The data is transmitted using ASK modulation in the UHF band at 433.92 MHz. Howeever, its output (14 dBm) is not enough to be transmitted over long distances. Therefore, an amplifier, MGA-30889, is used to provide up to 15 dB gain.

The beacon transmitter will be independent from the on-board computer and controlled by the EPS micro-controller. This will make the beacon more independent from the transceiver and more resilient even in the case of damage to the on-board computer.

Antenna Subsystem

The most important part in the subsystem is the antenna itself. For the design of an antenna, its directivity and bandwidth are also important. If we go for a very directive antenna, the CubeSat needs to be pointed accurately in order to have a high gain in the direction of interest. In CubeSat, the attitude control is made as simple as possible and, therefore, the resulting accuracy won't be very good. Thus, a very directive antenna is undesirable. The antenna configuration should also allow for a large enough bandwidth to accommodate the different signals with sufficient guard band to avoid interferences. The dipole antennas fit perfectly these requirements. The wavelength of the antenna is around 70 cm and, therefore, the antennas must be at least a quarter of wavelength (around 17 cm). In order to avoid interferences, it is obvious that a deployment system must be designed and that the antennas should extend beyond the one side of the CubeSat as its largest dimension is only 10 cm. The antennas can be held together by a string while being launched. Afterwards, the material holding the antennas together can be burned to deploy the antennas in space. The antenna subsystem also consists of antenna conditioning circuits like baluns and power splitters for generating a circularly polarized wave. Moreover, it also consists of a signal combiner to combine signals from three different sources. For the power splitter and combiner, QCN-5D+ from Mini-circuits will be used.

The antenna has two crossed dipoles polarization. It is the longest deployable device with 17 cm long monopoles. The deploy system has been designed using a nylon cable that when molten, will detach them from the cubesat body. During the launching, it is important to keep the antennas (monopoles) folded, because of this, it is necessary to have a deployment system that deploys the antennas in the correct moment. This correct moment is more or less thirty minutes after the moment when the launcher puts the CubeSat into orbit. The part of the timing is controlled by OBC, but in these lines is only treated the hardware that allows OBC to deploy the antennas in the correct moment. The main idea of how to carry out the deployment is simple: a current circulates through a low power resistor which heats a lot and burns the nylon cable that is holding the folded antennas. When the nylon burns, it drops the antennas and they return to their original position (deployed). The first idea was to put the resistor in parallel with the communications subsystem (in the communications board) and when the communications subsystem turns on, the current circulates through the resistor, the antennas are deployed and the resistor becomes an open circuit to not affect the operation of the communications subsystem. This was a good idea because it takes advantage of the current that enables the communications for deploying the antennas. But the main disadvantage was that in the testing was discovered that most of the resistors do not become an open circuit, even using very low power resistors (1/12 W, 1/8 W…). This is a very important problem because if the resistor does not become an open circuit, and it shortcuts the communications board. Different deployment systems were investigated, and it was discovered that in most of the picosatellites, an independent system is used for the deployment.

Following the idea of an independent system in charge of burning the resistor, a PCB was designed by Nil Vernis (System Integration) using the integrated circuit LT1118, which can provide 1.2A and 6,5V. The idea is that this system is allocated in the EPS board and the current goes through the SatBus to the communications board where the resistor is allocated (but is independent of the comms board). The reserved pin for the antenna deployment is the A1. The value selected for the resistor was 10 ohms, because lower values do not heat such as high values, but high values limit the current, so there was a compromise between the heat and the current, and finally 10 ohms were chosen. The allocation of the resistor in the comms board was done by Sumit Karki, trying to put it as in the middle as possible, in order to make easier the pass of the nylon coming from the antennas.

In the next images you can see the resistor allocated in the comms board:

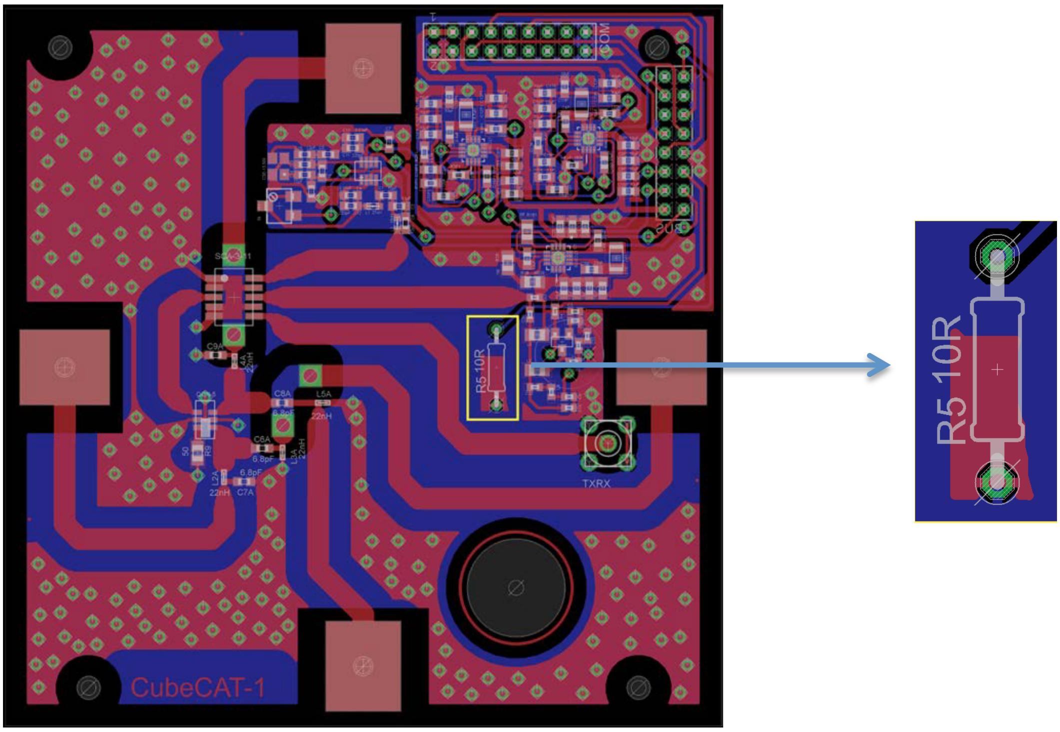


Figure 2.5 Antenna deployment burning resistor

2.1.4 Attitude system

The ADCS system is based on a mixture of active and passive systems. This design was chosen in order to ensure most of the success of the mission even if the active control fails. The passive control is needed to ensure the correct pointing for the communication. Furthermore, it doesn't need any power to work. On the other hand, the active system is needed to take pictures from the camera; it is considered for us as a technological demonstrator for further more complex 3D control systems. Its conceptual approach is shown in figure 8 and explained on the following sections.

Passive System

The passive system is divided in a magnet and two magnetic hysteresis rods to dissipate the energy of the oscillations. The constant magnetic momentum generated by the permanent magnet will orient CubeCat-1 structure parallel to the Earth's magnetic field (EMF). This system only guarantees that the passive magnets will be oriented in this certain way but it doesn't guarantee the angular position around the x-axis (the one parallel to the passive magnets) that is addressed in next section.

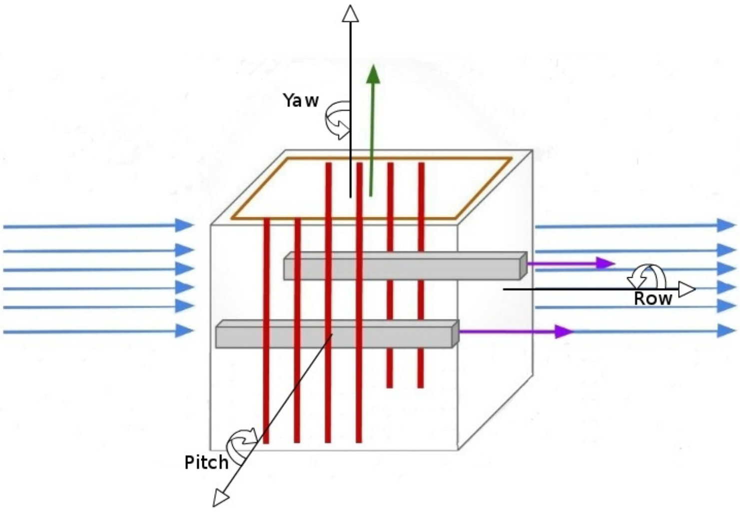


Figure 2.6 ADCS scheme

The problem with only a magnet is to stop the row oscillations; the magnet has no effect on its axis. This is the reason of using some metal with magnet hysteresis. The hysteresis rods working principle is based on the magnetic hysteresis phenomenon. The induction of the rods, caused by the interaction of the EMF with the material, is dissipated as friction inside the molecular structure of the metal (AD-MU-80 metal). Some simulations with passive magnets have been performed and have demonstrated that it stops the pitch oscillations after some hours and needs more time for the yaw. This graph is obtained experimentally with a cube structure hung on the ceiling and while the oscillation was being recorded with a camera with a camera its oscillation.

The obtained results show that the ones with the magnet have a higher oscillation frequency and stops after 30 minutes.

Active System

The active system is divided in determination, actuator and software part. To reach an approximation of the satellite attitude parameters it is necessary to obtain the information from different sensors. The determination system implemented in the CubeSat is based on the information obtained by a magnetometer, a gyroscope and accelerometers. The 9DOF Razor IMU gives all the sensors.

Magnetometer: Responsible of the magnetic field sensing. With the measures obtained and the knowledge of the EMF (Earth's Magnetic Field vector).

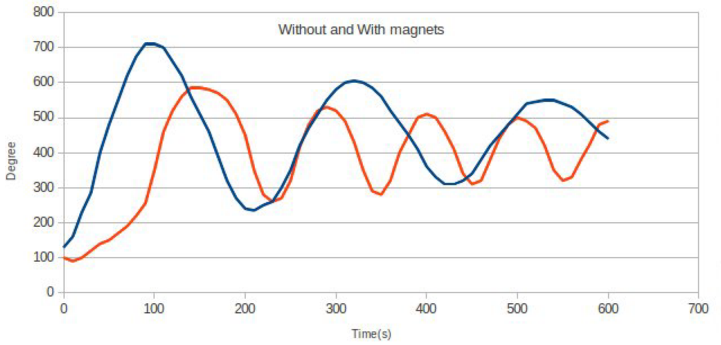


Figure 2.7 Test without (blue) and with magnet (red)

This will help to know the correct position of the satellite to the Earth magnetic field and know the relative position to earth surface.

Gyroscope: It measures the angular velocity around each axis (yaw ψ**, pitch θ** and roll φ). It will be used for sensing the oscillations and help with more data the active control software. The actuator decided to use in terms of consumption and less restrictive with the passive magnets is an active coil with a ferromagnetic core. The field vector generated during the activation of the coil will interact with the magnetic field of the magnet (see figure 10). The resultant vector from the vector sum of the ones created by the passive one and the coil will align with the EMF vector. Where Bc is the coil magnetic field, Bi magnet magnetic field and α degrees that the system is going to turn.

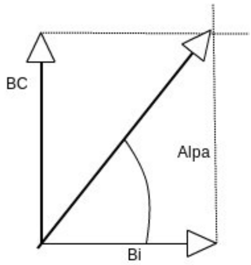


Figure 2.8 Magnetic fields

To turn α degrees the coil magnetic field and passive magnet field must satisfy:

$$\alpha = \tan^{-1}\left(\frac{B_i}{B_c}\right)$$

Where the coil magnetic field $B_c = \mu_r \mu_0 \frac{NAI}{L}$ is:

One difficult problem of this interaction is that the magnetic field generated by the active coil have to be higher then the passive magnets, to ensure that the satellite can make a turn of approximately 90°. The coil drive circuit has a full bridge to control the current direction and a current sensor to test if the circuit works correctly. The current of the coil will be characterized in front of duty cycle for avoiding a control loop.

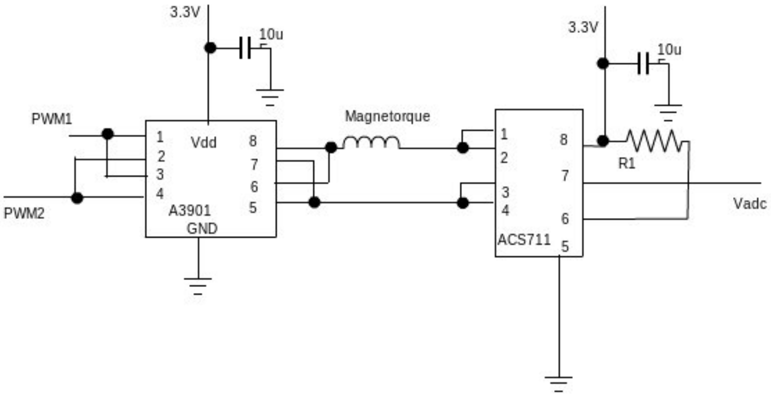


Figure 2.9 Coil circuit

The active software has two different controls. The fastest one that uses the oscillations of the satellite to take the photo and the other one that waits until it stabilizes. One of the PWM modules of the OBC is in charge of controlling the mean current that flows through the coil.

2.2 PAYLOADS

As mentioned in the objectives of the project, CubeCat-1 mission has 11 different experiments and technology demonstrators to be tested. While the resources required are considered in the budgets section, their description is shown briefly in the following list:

- **Wireless Power Transfer experiment** to be done under radiation conditions. It is a partial part of Elisenda Bou PhD thesis on satellite formation flying. While Wireless Power Transfer will be demonstrated on board of the ISS by the end of this year, the effects of near-field radiation (present in free-space) upon a wireless power transfer link have yet to be analysed. This payload will be a technology demonstrator of Resonant Inductive Coupling Wireless Power Transfer in free-space and also give the community valuable data about the effects of near field on WPT systems, which will be a decisive factor for the practical deployability of in-space WPT.

  This system is the second deployable allocated in the satellite. The WPT experiment consists of two coils that need to be placed outside the satellite. As happens with the antennas, they need to be attached to the cubesat structure during the launch, so a deployment system is needed. Mario Gómez and myself designed this system but it is not definitely designed and tested. The design of this deployable consists of a cable attached to the end of the coils that when burned, instead of melting, it shrinks and pulls the coils to their deployed position.

- **MEMS monoatomic oxygen detector**, designed and manufactured by UPC it is going to study the presence of this chemical molecule in the lower parts of the ionosphere.

- **Peltier fed autonomous beacon emitter**, to be used experimentally for energy harvesting purposes, it is going to be fully independent from any other subsystem on the satellite.

- **CelSat Si solar panels**, designed and manufactured by UPC to study the degradation of this design under true space conditions.

- **New topology transistor**, never tested in space before.

- **Geiger counter** to measure Beta radiation levels; needed to correlate the radiation with the results of the previous experiments.

- **Digital CMOS camera** to obtain images from the Earth. It has VGA resolution and jpeg compression to ensure the capability to transmit its data to the Ground Station. The main objectives are to obtain images from the Earth's surface from space while solving the complex problem of a whole satellite mission.

- **Other technology demonstrators**: COTS MEMS Inertial Measurement Unit, UPC designed EPS, attitude determination based on photodiodes, attitude control based on a permanent magnet, mu-metal bars, and a coil to control the orientation of the satellite

Bibliography (Chapter 2)

[1] www.cubesat.org

[2]Taskit PortuxG20 specifications http://www.taskit.de/en/products/portuxg20/index.htm

[3] PAE Cubesat, Mission Analysis

[4] PAE Cubesat, Orbit Parameters

[5] PAE Cubesat, Operating States

[6] PAE Cubesat, OBC

[7] PAE Cubesat, Communications

[8] PAE Cubesat, Payload

[9] PAE Cubesat, Attitude System

[10] PAE Cubesat, EPS

[11] Adriano Camps Carmona, Roger Jovè Casulleras, Elisenda Bou Balust, Eduard Alarcon Cot, and Juan Ramos Castro, *"FYS - CubeSat Proposal - CubeCat-1",* 2013.

[12] "Small Satellite Ground Stations", **Brochure, ISIS, 2013**.

CHAPTER 3: MISSION ANALYSIS

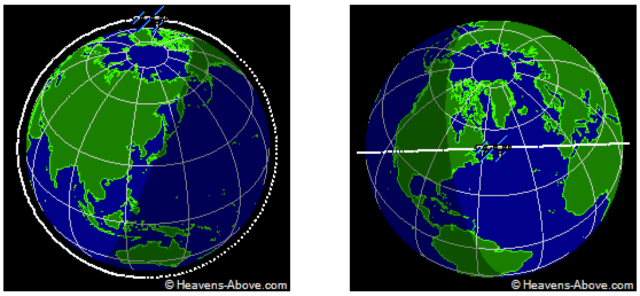
Mission analysis is directly linked to the work done in system integration. The results taken from this part are the inputs for the thermal analysis and also for the energy manager. This analysis is also important for some of the payloads as they are going to degradate and the 'how' and 'when' will depend on the orbit conditions. At the end of the report, the viability of the mission is going to be analised, based on the information taken from the different budgets and from the orbit conditions.

3.1 ORBIT ANALYSIS

3.1.1 Orbit parameters

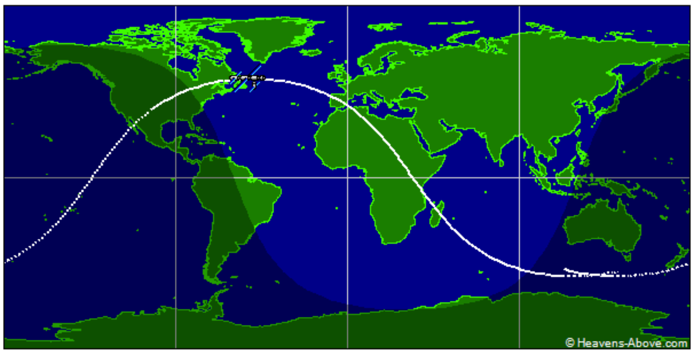
The considered orbit until now has been the one corresponding to the International Space Station (ISS) as we applied for the Flight Your Satellite (FYS) program from the European Space Agency (ESA).

The ISS orbits the Earth at 51.6° to the Equator, following the direction of the Earth's rotation from west to east. The Earth itself is tilted at 23.4° to the plane of its orbit around the sun (sun vector), so the ISS is orbiting at 75° to the sun vector. The ISS's altitude varies between 320 to 410 km, and it takes 92 minutes to circle the Earth. The orbit inclination offers good coverage of most of Earth's surface. As it takes about 90 minutes for the satellite to complete one full circuit, when it gets back to its starting point, Earth's surface has moved eastward. It moves 360 degrees in 24 hours, or 15 degrees per hour. So after each satellite circuit of 1.5 hours, the Earth's surface has moved about 1.5 times 15 degrees, or 22.5 degrees, further to the east. [2]



View from above orbital plane

View from above satellite



Ground track

Figure 3.1 ISS orbit

The parameters to determinate the orbit, are the following and they are given by [1]

```
Satellite: ISS
Catalog Number: 25544
Epoch time:       13175.71455703    =    yrday.fracday
Element set:      900
Inclination:        51.6474  deg
RA of node:         71.0334  deg
Eccentricity:      .0008360
Arg of perigee:     91.5026  deg
Mean anomaly:      268.7082  deg
Mean motion:     15.50371183  rev/day
Decay rate:      1.67170E-04  rev/day^2
Epoch rev:             3587
Checksum:               297
```

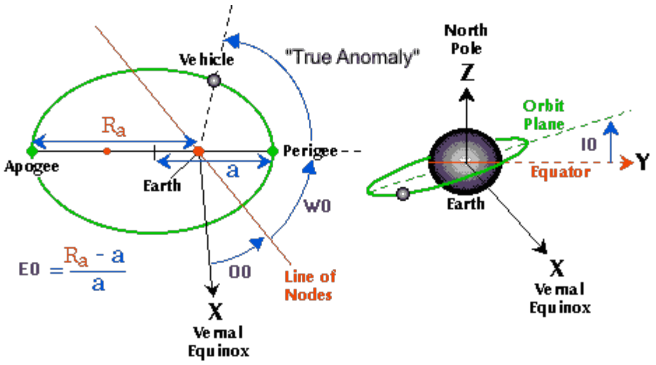


Figure 3.2 Orbit parameters [2]

They may vary depending on the date checking the data as it might suffer some deviations.

3.1.2 STK results

STK software is used to analyse space systems to know the conditions that will suffer the satellite when in orbit. The output data extracted can give us many types of information, but the most important in our mission and that will be used as input for other analysis is the sun radiation in each point of the orbit as it provides the energy input for the batteries.

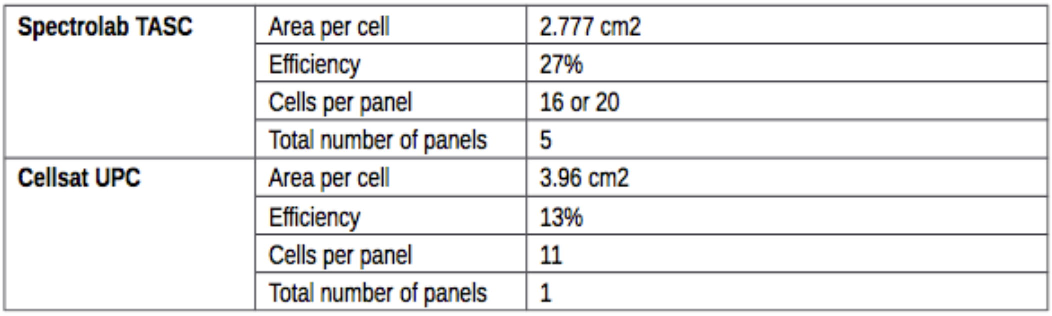
One of the best characteristics of the STK is that it allows you to analyse the conditions for your own satellite once you have the 3D model (section 4.2.5). Although it is a great point, it became very challenging moving our satellite model from Solidworks to a readable format for STK.

3.1.2.1 Solar Power

The solar intensity variation was computed using the AGI STK. Some assumptions were considered:

- The Solar Panel Power tool from AGI was used. This tool requires a model of the solar panels on the CubeSat, including their area and efficiencies. Therefore, the 6 panels of the CubeSat were modeled taking into account that there are two different types of cells (listed in table 3.1). A time step of 60 s was used.
- The attitude of the satellite was simulated using the Smart Nanosatellite Attitude Propagators (SNAP) tool from the Space Systems Laboratory of the University of Kentucky. As a first approximation, it was interesting to have different realizations of the satellite's attitude. The code was then slightly modified to introduce some randomness on the attitude files generated because the ones used were too deterministic. The parameter considered to be random was the initial rotation speed in the three axes. However, this did not perform as expected and this approach was abandoned due to the long time needed to obtain the attitude files.

The next step is to import the model in STK and define the properties for the solar panels. The effective area takes into account the number of panels facing the Sun for a particular attitude. The Solar Power Panel computes this effective area by projecting the sun vector on the satellite, which in turns varies depending on the attitude.



Spectrolab TASC	Area per cell	2.777 cm2
	Efficiency	27%
	Cells per panel	16 or 20
	Total number of panels	5
Cellsat UPC	Area per cell	3.96 cm2
	Efficiency	13%
	Cells per panel	11
	Total number of panels	1

Table 3.1 Solar panels properties

Using the Solar Panel Power tool, the solar intensity is computed. This solar intensity is a value between 0 and 100. It will be 100 when the satellite is completely illuminated by sunlight and 0 when in an eclipse. The solar intensity can be used as a factor when computing the power of a solar cell according to:

$$P = SI*A*Etha*1358$$

- P is the total available solar power.
- Etha is the solar panel efficiency.
- SI is the solar intensity.
- A is the effective area.
- 1358 W/m2 is the available solar density power in the Earth orbit.

This data was saved in a .csv format for convenience and later post-processing. Figure 3.3 shows the solar power received after processing the output data from STK. [3]

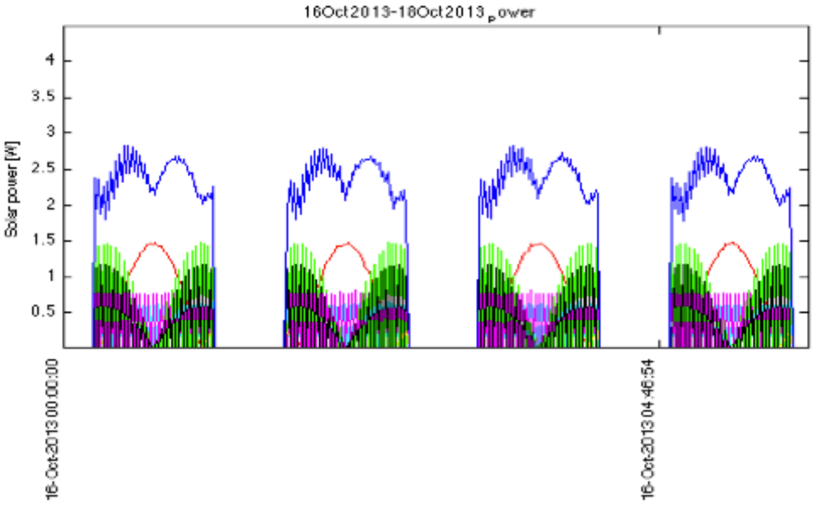


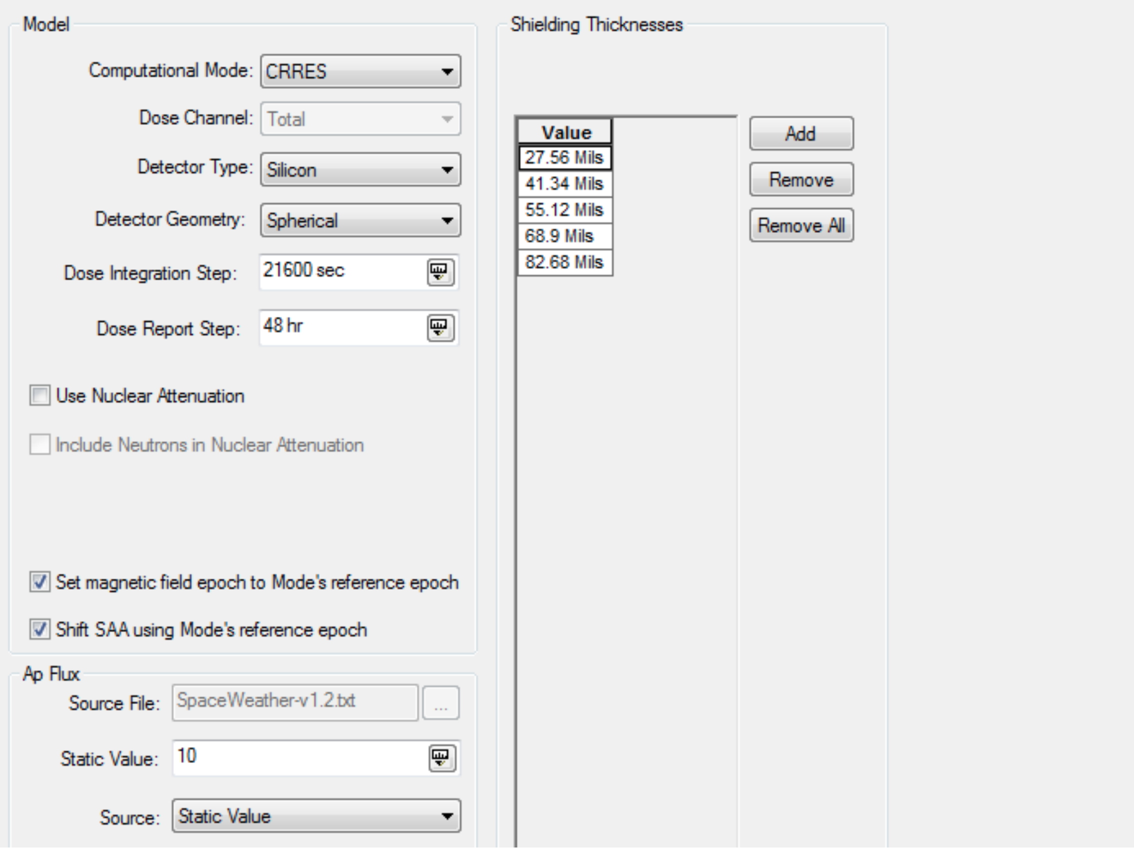
Figure 3.3. Solar panel power simulation for 4 orbits (about 6 hours)

3.1.2.2 Radiation analysis

This section contains a brief radiation analysis performed using the STK Space Environment and Effects Tool (SEET) from AGI. This add-on allows performing a radiation analysis from different available mode and radiation flux databases. There is a self-contained explanation with all the details on the SEET help dialog. [4]

For our purposes, it was interesting to obtain the accumulated radiation dose on the CubeSat for a given period and given the actual shielding thickness. It is clear that the space environment is a hazardous environment and the performance of the systems is affected due to these high-energy, charged particles (mainly ions). However, shielding can reduce significantly this particle flux and impacts. The thickness of this shielding must be then determined. The SEET considers aluminum (Al) as shielding material. In addition, the thickness will depend on the "detector type", i.e., the material under radiation. Here only silicon was considered as long as it is the main component on the electronics and subsystems of the satellite.

The next figure shows the SEET dialog with the parameters set for the analysis:



Model

Computational Mode: CRRES

Dose Channel: Total

Detector Type: Silicon

Detector Geometry: Spherical

Dose Integration Step: 21600 sec

Dose Report Step: 48 hr

☐ Use Nuclear Attenuation

☐ Include Neutrons in Nuclear Attenuation

☑ Set magnetic field epoch to Mode's reference epoch

☑ Shift SAA using Mode's reference epoch

Ap Flux

Source File: SpaceWeather-v1.2.txt

Static Value: 10

Source: Static Value

Shielding Thicknesses

Value
27.56 Mils
41.34 Mils
55.12 Mils
68.9 Mils
82.68 Mils

Figure 3.4 SEET parameters

The CRRES mode was selected. This flux model is based on data collected by the Combined Release and Radiation Effects Satellite mission [4]. Data from both electron and proton particle energies is available (i.e., a combined dose report can be obtained). The detector type was set to silicon and the detector geometry to an omnidirectional irradiated sphere. This geometry suits more properly the actual CubeSat geometry because the electronics are contained inside the satellite and the attitude will vary according to the magnetic field and thus the irradiated source direction. The detector type is embedded at the center of it.

The simulation is performed to find the suitable thickness for a lifetime of one year. The result is shown in Figure 3.5. In this case, the dose integration step was set to 1.5 hours.

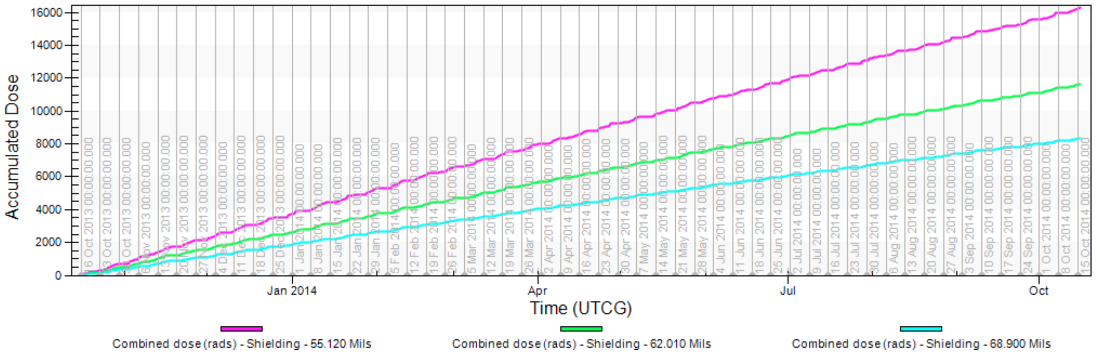


Figure 3.5. Combined accumulated dose (rads) for different shielding thickness

Using these results, the kind of shielding that will be used in the mission has beeen decided applying a 10krad threshold and its implications are added to the overall design specifications.

Aluminum density	2.8 g/cm3
Shielding thickness	62.01 Mil (1.57 mm)
Extra shielding board sizes	10x10x(t-0.07) cm3
Sides to shield	6
Extra weight	146.2 g

3.1.2.3 Access opportunities

When analysing the orbit, another item to take into account is the communication between the satellite and the ground station in Barcelona and the other way round. The STK software provides us the information to know when the communication can be accomplished.

The next table contains the scenario objects for the access opportunities:

Satellite	Name:	CUBECAT
	Altitude:	650 km
	LTAN:	06:00:00 UTCG for OrbitAnalysis06 scenario. 12:00:00 UTCG for OrbitAnalysis12 scenario.
Facility	Name:	Barcelona
	Coordinates:	Latitude: 41.3833° Longitude: 2.18333°
Transmitter	Frequency:	433 MHz
	EIRP:	30 dBm
	Polarization:	Right-hand circular
	Data rate:	1200 bps
	Modulation type:	FSK
Receiver	Frequency:	433 MHz

	G/T:	20 dB/K
	Polarization:	Right-hand circular
Antenna	Antenna type:	Dipole
	Frequency:	433 MHz
	Length:	34 cm
Sensor	Name:	CCD_camera
	Sensor type:	Simple Conic
	Cone Angle:	28º (based on focal distance and satellite altitude)
	Elevation:	90º

Table 3.2 Access parameters

The satellite-to-ground station distance during access opportunities is computed for link budget purposes. A unique LTAN (Local Time Ascending Node) was considered because there is no dependency on it, except for some particular year periods.
The access opportunities are crucial to download the payload data and upload the telemetry commands.

The values of interest are placed in the next table:

```
Global Statistics
-----------------
Min Duration       1070    15 Oct 2014 02:50:16.961    15 Oct 2014 02:50:22.820            5.858
Max Duration        688    11 Aug 2014 08:43:44.270    11 Aug 2014 08:50:29.387          405.116
Mean Duration                                                                            322.906

Total Duration                                                                        692956.955
```

Table 3.3 Access opportunities

The next figure shows the access possibilities for 4 days starting the 16th of April 2014. Using four days is only to show the results as it is enough to see the periodicity of the access possibilities.

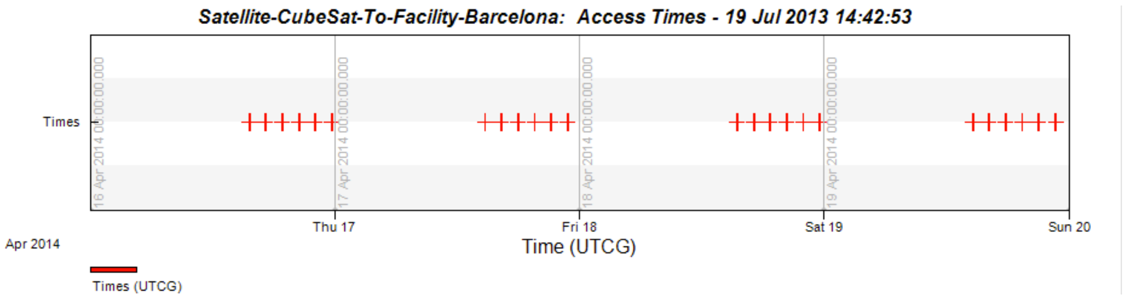


Figure 3.6. Access opportunities

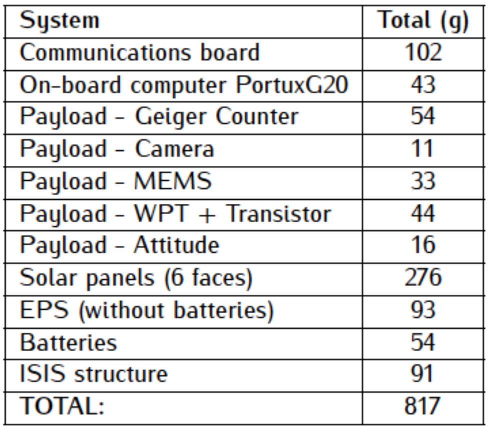
The information in the plot above together with the results of the budgets in the next sections determine the viability of the mission and when to download information or upload new orbit parameters to the scheduler.

3.2 BUDGETS

The next two reports are the ones that show if a mission will be successful or not. The mass budget gives an approximation of how heavy our cubesat will be. If it exceeds the maximum weight we will probably have to discard some of the payloads. On the other hand, the power budget gives us information about the consumption of every subsystem and payload of the cubesat. This power consumption is used in the energy manager simulator, which predicts if the satellite will run out of power and therefore the mission is unsuitable.

3.2.1 Mass budget

The CubeSat based standard limits the mass of the satellite to a maximum weight of 1.33kg. In CubeCat-1 analysis it is desired to use less than 1kg of mass in order to have 300g as safety margin to equilibrate the centre of gravity to a 1cm radius sphere around the geometrical centre required by the launch. The mass of the different devices were estimated due to the biggest parts and are presented in table 3.2.



System	Total (g)
Communications board	102
On-board computer PortuxG20	43
Payload - Geiger Counter	54
Payload - Camera	11
Payload - MEMS	33
Payload - WPT + Transistor	44
Payload - Attitude	16
Solar panels (6 faces)	276
EPS (without batteries)	93
Batteries	54
ISIS structure	91
TOTAL:	817

Table 3.2 Mass budget

This information is not only to compute the total weight of the cube, it must also be used to place the components in the right place to have the centre of mass not too much displaced from the real center of the satellite. This need to have the mass center within a radious of one centimetres from the real center, establishes the position of the heaviest components such the camera, the batteries or the attitude coils. The determination of the center of mass and its implications are explained in the next chapter as they are used to design the differents PCBs and to place the mentioned components in the right place.

3.2.2 Power budget

The power budget must contain both power generation and power consumptions values. As the power generated is obtained using STK, the table shows the power consumptions of each device, how long it is on, and its frequency of operation.

SOC (State Of Charge)

The energy input and storage is defined by the power received from the sun in the solar panels and its storage in the batteries. The next image shows the profile of the energy that gets into the satellite and stays in it if there are no payloads operating; wich is called the SOC of the satellite.

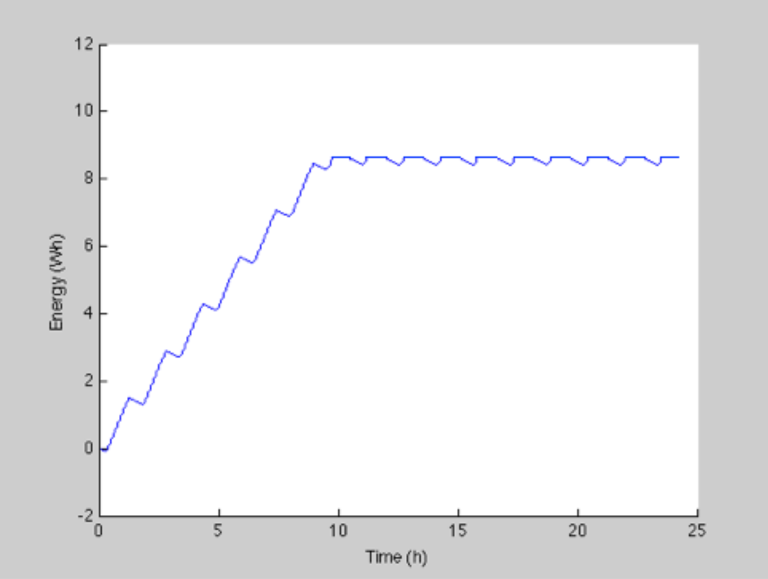


Figure 3.4. Energy SOC

The first part of the line is the time while the batteries are charging and the second piece plots the energy in the batteries when charged taking into account that there is a permanent consumption due to the subsystems.

Energy output

The next table shows the power consumption of each device as also their operation time. The lines after the table describe the energy states of the satellite and the power profiles of each subsystem.

System	Power (mW)	Frequency	Duration
Communications board Tx	3000	orbit dependent	10*min*
Communications board Rx	96	all time	
Beacon transmitter (433 MHz) – WRL-10534	165	every 2*min*	few secons
Mixer – LT5512	340	every 2*min*	few secons
Local oscillator – ASVMPLV-BLANK-XR-T	120	every 2*min*	few secons
Communications amplifier (beacon) – MGA-30889	385	every 2*min*	few secons
Electrical Power Subsystem	100	all time	
On-board computer PortuxG20 (on)	450	sw programmed	
On-board computer PortuxG20 (sleep mode)	50	sw programmed	
Payload – Geiger Counter	150	10*min*	10*min*
Payload – Camera	15	1x day	5s
Payload – MEMS	100	1 x day	2*min*
Payload – Grafè	175	1 x day	2*min*
Payload – WPT	300	every 2 weeks	10*ms*
Payload – Attitude	280	on demand	30*min*

Table 3.3 Power consumption

Later on, some modifications explained in section 4.3.2 were made reducing the OBC power consumption to the half.

The next sections also shows the power profiles of every subsystem and gives a detailed explanation of which parts need to be turned on when carrying out an experiment. This explanation is based on the possible states of the satellite during the mission shown in the next image:

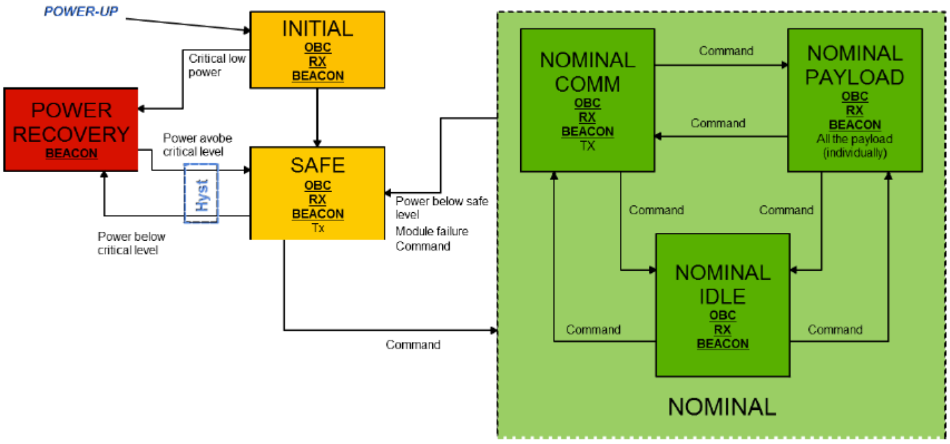


Figure 3.5. Energy states (by Adrià Amezaga)

The diagram above shows the diagram of the operating states. Triggers are next to the arrows that describe the direction of state transitions. Each box contains the name of the state in capital letters and below that, the systems that must be turned on all the time in underlined bold and the systems that can be turned on.

Initial: When the system powers-up for the first time, the OBC checks the state of all the systems performing actual measurements. The results of the checking are stored in a file for later transmission. After the check the system enters to the SAFE mode.

$\mathbf{P_{INITIALmax} = P_{OBC} + P_{RX} + P_{BEACON}}$

- OBC: 250 mW, all the time
- EPS: 100 mW, all time
- RX
  - Comm board reception: 55.5 mW
  - Reception LNA: 21 mW
- BEACON:
  - Beacon transmitter: 165 mW
  - Mixer: 340 mW
  - Amplifier: 385 mW
  - Local oscillator: 120 mW

$P_{INITIALmax} = 1{,}4365$ W

Safe: After all checks in the INITIAL operating mode are done, the system enters this state. Only the OBC, the receiver and the beacon are permanently active. Transmitter can be activated with a command in order to transmit the results of the checklist or some other information by demand. The system can enter into the SAFE mode from POWER RECOVERY if the level of energy is above the critical level, or from NOMINAL mode if the power is below a safe level, there is a module failure or by means of a command.

$\mathbf{P_{SAFEmax} = P_{INITIAL} + P_{TX}}$

- TX
  - Comm board transmition: 99 mW
  - Tx amplifier: 3000 mW
  - Mixer: 340 mW
  - Local oscillator (same as beacon)

$P_{SAFEmax} = 4{,}8755$ W

Power recovery: The system enters into this state only if the energy level is below a critical level. All the systems except the beacon are inactive. This means that commands can't be received but the battery state of charge and other parameters (yet undefined) can be received from the ground station. The only way to get out of this state is an energy level above a critical level.

Nominal modes: The system enters this state only if a command is received when the SAFE mode is active. There are nominal sub-states (IDLE, COMM and PAYLOAD). This subdivision allows more control on the tasks being performed by the satellite. These profiles are estimated using the design specifications of each subsystem and payload.

- COMM: This mode is intended to be used only when data has to be shared between the satellite and the ground station.
  $\mathbf{P}_{\mathbf{COMM}} = P_{TX} = 3{,}439$ W

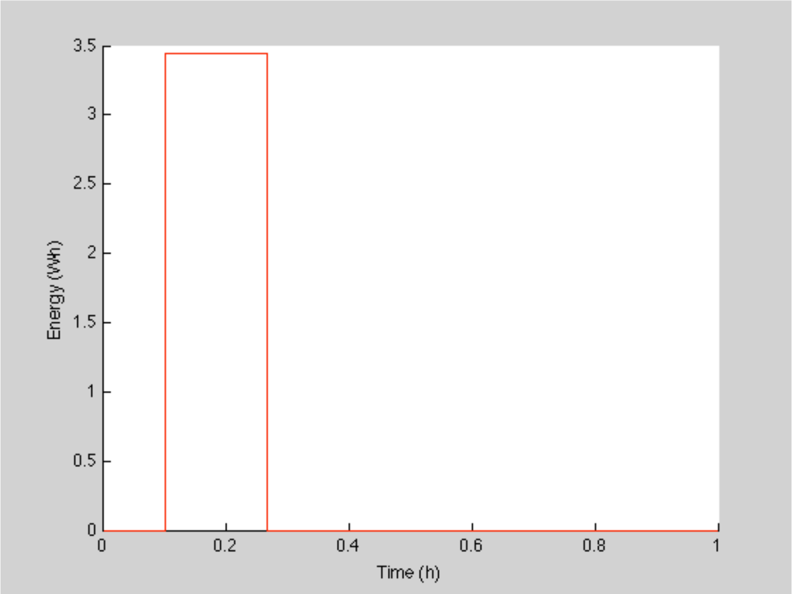


Figure 3.6 Tx power consumption profile

- IDLE: Used for other purposes, mainly when there is no science to do and a conservative usage of power is wanted.
  $\mathbf{P}_{\mathbf{IDLE}} = P_{INITIAL} = 1{,}4365$ W

- PAYLOAD: For science exclusively. This is the state that will be active most of the time. In this case, the consumption profile of each payload is shown so it is easier to compare between them, so the consumption is not only a matter of power but also of time.

$P_{ATTITUDEmax} = 280$ mW
$P_{CAMERAmax} = P_{ATTITUDEmax} + 15$ mW $= 295$ mW

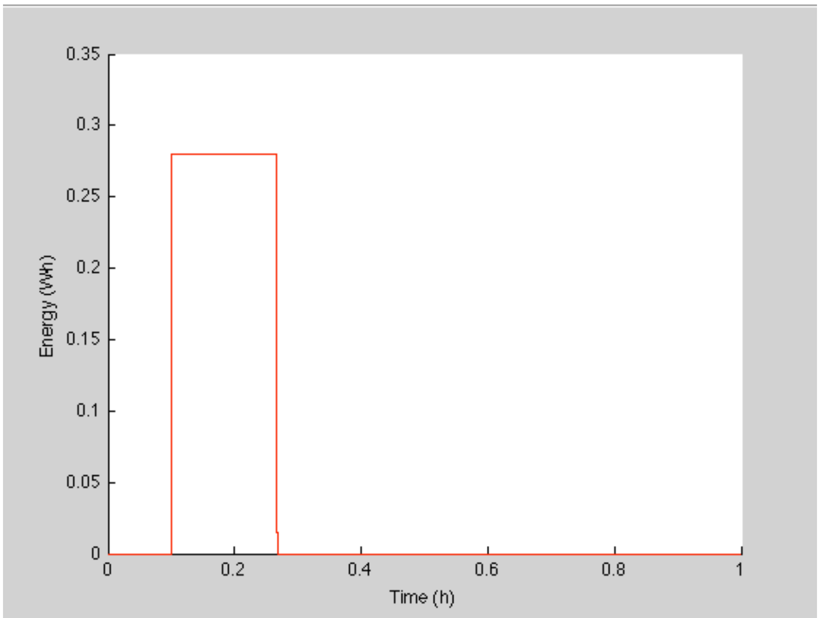


Figure 3.7. Attitude power consumption profile

$P_{WPT} = 300$ mW

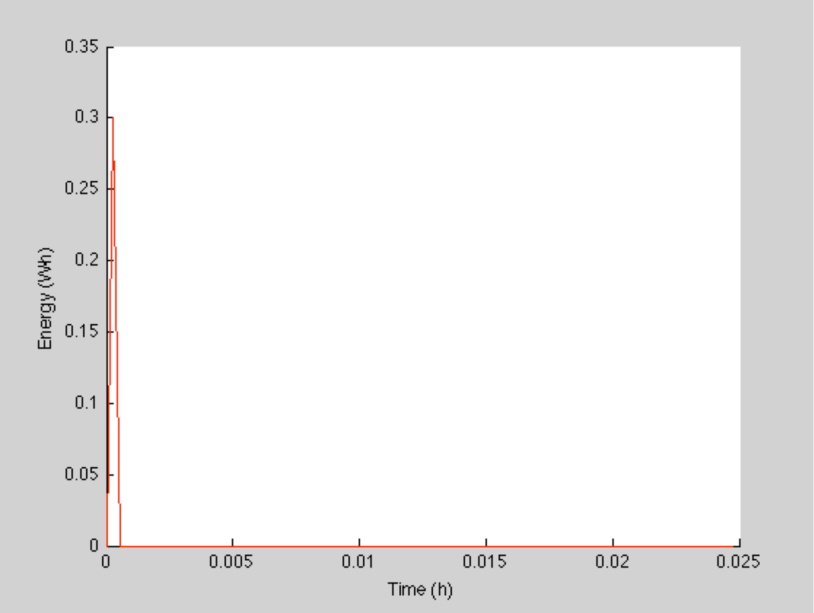


Figure 3.8. WPT power consumption profile

$P_{GRAPHENE} = 175mW$

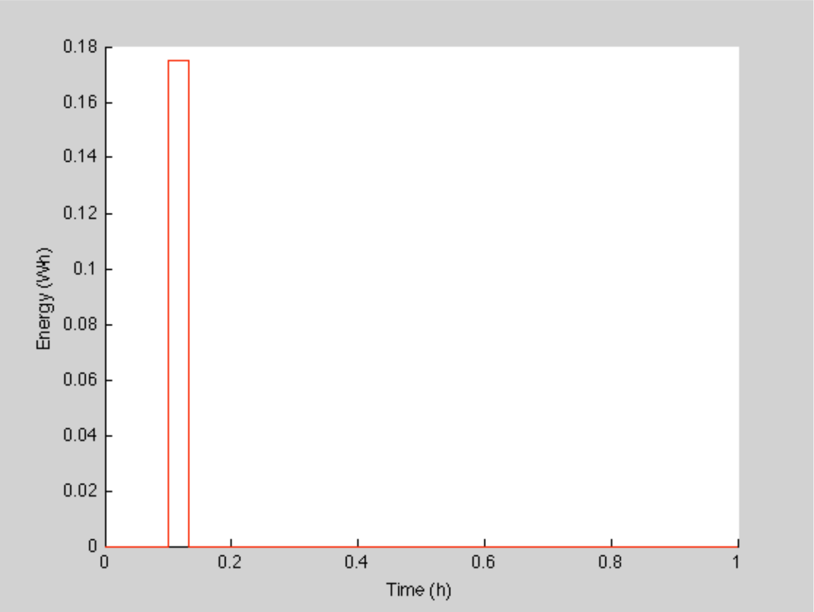


Figure 3.9. Graphene power consumption profile

$P_{MEMSmax} = 100\ mW$

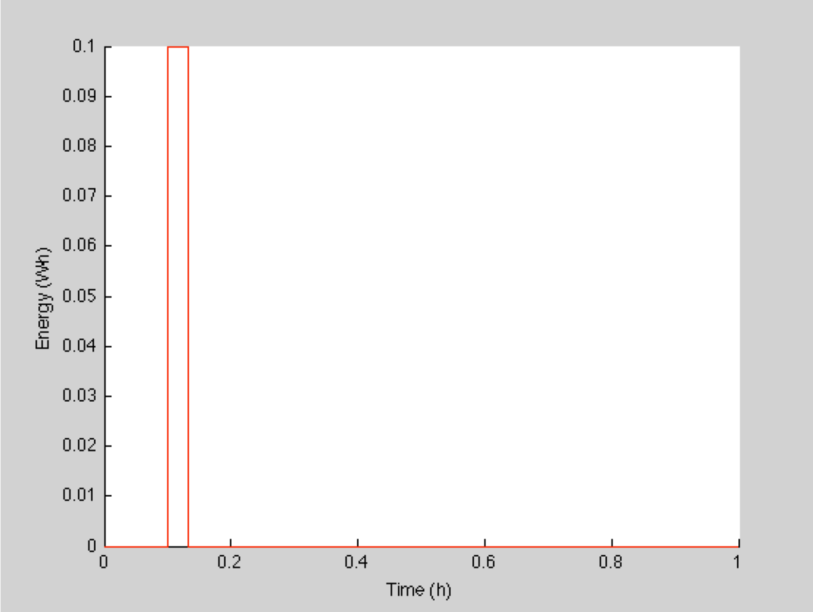


Figure 3.10. MEMS power consumption profile

Having these power profiles, it is important to be careful with the transmitions, as they need a lot of power and could make the satellite run out of it causing the lost of some data and having to wait when facing the sun to start operating again.

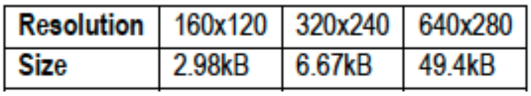
This will be a very important point at the end of this report when the viability of the mission will be analised.

3.2.3 Data budget

The aim of this study is to give detailed information for the program that computes the amount of data generated by the payloads. The motivation of doing this study is to know the size of the information that will have to be sent and the viability of sending all of it and to know if all this data can be stored on the satellite too.. This is the last budget that is needed to plan the mission. The information taken from this part must be combined with the power budget and with the information of the access opportunities and decide when to download or upload information.

The next lines give information about the amount of data generated by the payloads and are another input for the viability of this mission considered in the next chapter.

- CAMERA: this payload can take pictures in 3 different resolutions and with different compression ratio (Table 3.4). In this document there is only the information about the size of the pictures without applying any compression ratio because it is not known which one will be applied (this document will have to be updated as soon as it will be known, however it has a random component of the compression because it depends on the different colors of the image).



Resolution	160x120	320x240	640x280
Size	2.98kB	6.67kB	49.4kB

Table 3.4. Resolution and data size

- WPT: However this experiment is really configurable and has a wide range of possibilities that lead to different volumes of data, by default it will generate 2^{16} packages. Each of these packages will contain up to 24 bits because it will be used the same ADC as in the graphene experiment.

- GRAPHENE TRANSISTOR: the graphene transistor will generate two voltages: Vg and Vd with 12 bits of resolution each DAC (Digital to Analog Convertor). The measurements are done with an ADC of 24 bits. This leads to a data volume that can go from 0 to 50MB. It can be assumed that at the beginning the data amount will be around 72kB, 8 bits for Vg and Vd and 9 bits for lds.

- MEMs: in this experiment, to do a complete measure it has to get 3 modes. Each mode has the size of 488 bytes so in total 1464 bytes have to be sent. However the first 8 bytes are not reliable and the other 480 are averaged every 48 bytes so as a result of this averaging we have a data volume of 16 bytes per mode. So counting that we have to send 3 modes and the date, time and temperature of the measure, the data that will have to be sent is 60 bytes per measurement.

- GEIGER COUNTER: this payload will be generating information during a whole orbit, so the volume of information that will be generated will depend on the number of samples desired to get in each orbit. Each sample has the volume of 16 bytes.

- MONITORING: this is the data generated by the sensors and that will be transmitted to the ground station. This data will be stored in registers. Here can be distinguished two particular cases: if only one register is going to be sent or if more than one register will be sent. This data generated is 54 bytes every 10 seconds.

3.3 THERMAL ANALYSIS

It is essential to perform a thermal analysis in order to determine the operation temperature at which the satellite will be once in orbit. This simulation uses the cubesat's 3D model build using Solidworks explained in section 4.2.4. The program used to do the thermal analysis is the Thermal Desktop, an AutoCAD plug-in. It allows you to simulate orbits, give properties to every material and post processing the results to have more accurate information.

This thermal analysis took about 3 month to get completely performed due to the lack of knowledge about thermal metrics and about the Thermal Desktop software.

3.3.1 Thermodynamics theory

The heat balance dynamics were used to obtain an expression that conduces to a useful result; the used expression is the following, which relates the input/output power with the temperature variations:

$$Q_{in} - Q_{out} = m \cdot C \frac{dT}{dt}$$

Some parameters are needed before computing the result of this expression, which mainly are the input and output heat energy. As in the space there are no convection, only conduction and radiation modes are present.

The absorbed heat corresponds to the direct solar radiation, the earth's albedo (the part of solar radiation that is reflected on the earth and arrives to the satellite), the earthshine, and the dissipated power inside the spacecraft, which contributes to heat it (by conduction). It results from the following equation:

$$Qin = P_{sun} \cdot S_{il.\ sun} \cdot \alpha + P_{albedo} \cdot S_{albedo} \cdot \alpha + P_{earthshine} \cdot S_{il.\ earth} \cdot \varepsilon + P_{disip}$$

The black-body is an idealized object that absorbs all radiation coming from any direction and wavelength, and emits isotropically in any direction. Its radiation only depends on its temperature, but the real body can absorb and emit only with a scaling factor called absorptivity and emissivity (α, ε). These parameters depend on the direction and the wavelength, and for a given direction and directional spectrum, the absorptivity equals the emissivity. They have a big dependence with the wavelength, so they are called (α) when the radiation comes from the sun (visible spectrum), and (ε) when it is leaving the spacecraft (also when it is coming from the earth, infrared spectrum). The emitted heat is the part of radiated power due to the temperature of the body object of the study, following the Stephan Boltzmann's law. Its radiation is characterized by:

$$Qout = S \cdot \varepsilon \cdot \sigma \cdot T^4$$

All this theory is included in the Thermal Desktop software that is the one used to obtain the thermal analysis for the mission, which is explained in the next sections.

3.3.2 Thermal model

When designing the model for the thermal analysis, it is important to build every part as a surface as at first they were built as solids and the simulations weren't unable to work properly. The model includes many different layers as each group of parts or materials of the cube have different properties both thermophysical and optical. The different layers in which the parts are separated are the structure, the sides of the cube, the solar panels, the antennas, the PCBs of the different floors and the deployables.

The aspect of the satellite once modelled is shown in the next picture.

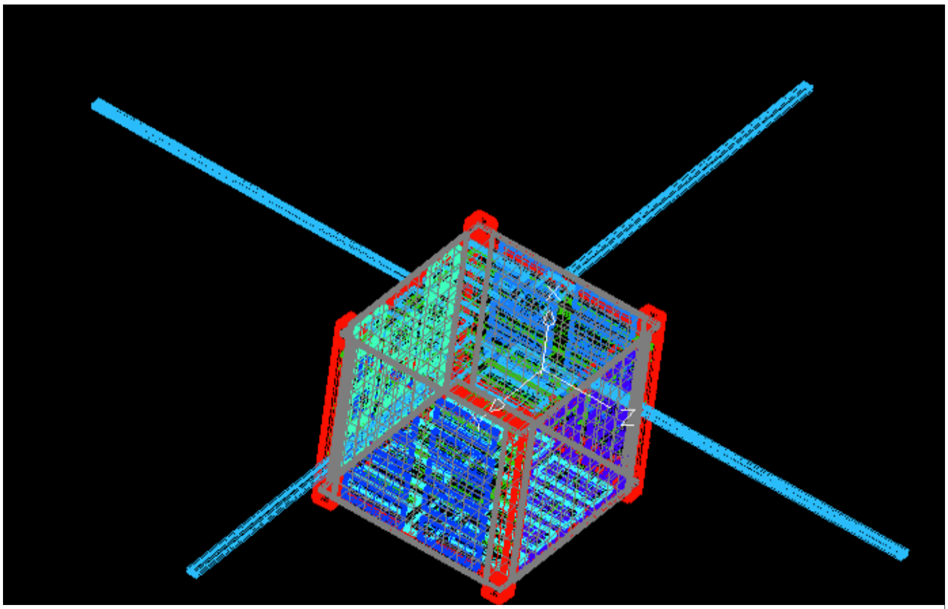


Figure 3.3 Thermal model

3.3.3 Material properties

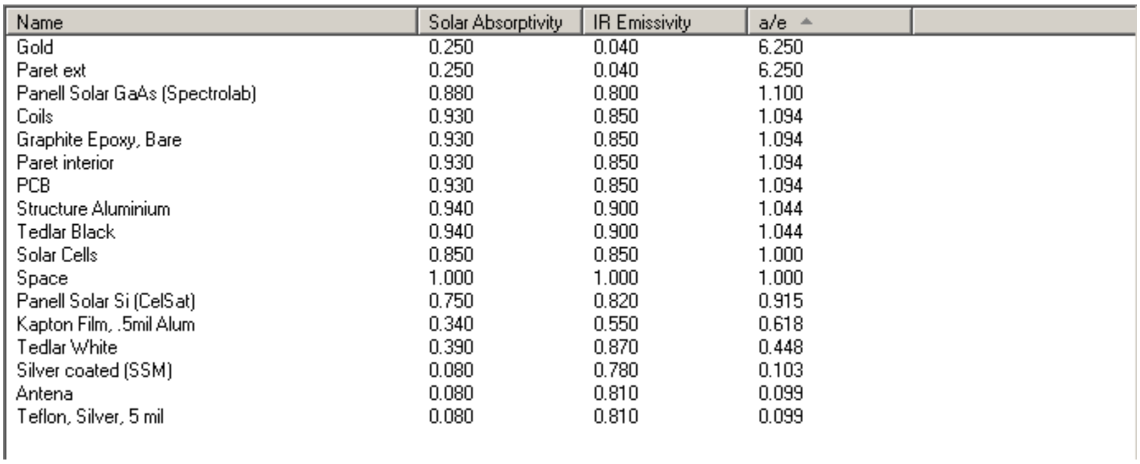
Once the model is finished the properties of each group of materials are given.

The properties in the Thermal desktop are divided in two parts, optical and thermophysical.

3.3.3.1 Optical properties

By "optical property" is meant a material's response to exposure to electromagnetic radiation and, in particular, to heat transfer. The most important optical properties of a material are its internal and external transmittances, surface reflectance, and refractive indexes.

The considered properties in the analyser, are the solar absorptivity the infrared emissivity and the a/e ratio between them. The absorption coefficient determines how far into a material light of a particular wavelength can penetrate before it is absorbed. The emissivity of a material is the relative ability of its surface to emit energy by radiation. It is the ratio of energy radiated by a particular material to energy radiated by a black body at the same temperature. The next image contains the different material defined for the satellite and their optical properties.



Name	Solar Absorptivity	IR Emissivity	a/e
Gold	0.250	0.040	6.250
Paret ext	0.250	0.040	6.250
Panell Solar GaAs (Spectrolab)	0.880	0.800	1.100
Coils	0.930	0.850	1.094
Graphite Epoxy, Bare	0.930	0.850	1.094
Paret interior	0.930	0.850	1.094
PCB	0.930	0.850	1.094
Structure Aluminium	0.940	0.900	1.044
Tedlar Black	0.940	0.900	1.044
Solar Cells	0.850	0.850	1.000
Space	1.000	1.000	1.000
Panell Solar Si (CelSat)	0.750	0.820	0.915
Kapton Film, .5mil Alum	0.340	0.550	0.618
Tedlar White	0.390	0.870	0.448
Silver coated (SSM)	0.080	0.780	0.103
Antena	0.080	0.810	0.099
Teflon, Silver, 5 mil	0.080	0.810	0.099

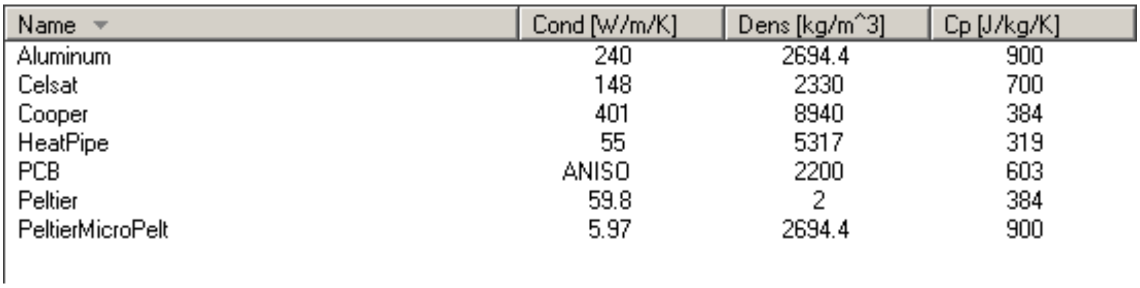
Figure 3.4 Optical properties

3.3.3.2 Thermophysical properties

Thermophysical properties are all material properties affecting the transfer and storage of heat, which vary with the state variables temperature, pressure and composition (in mixtures), and of other relevant variables, without altering the material's chemical identity. These properties will include thermal conductivity and diffusivity, heat capacity, thermal expansion and thermal irradiative properties, as well as viscosity and mass and thermal diffusion coefficients, speed of sound, surface and interfacial tension in fluids.

The thermophysical properties required for the analysis are the thermal conductivity, the density and the specific heat. Thermal conductivity is the intrinsic property of a material, which relates its ability to conduct heat. Heat transfer by conduction involves transfer of energy within a material without any motion of the material as a whole and the specific heat is defined as the amount of heat per unit mass required to raise the temperature by one degree.

The next image contains the different material defined for the satellite and their thermophysical properties.

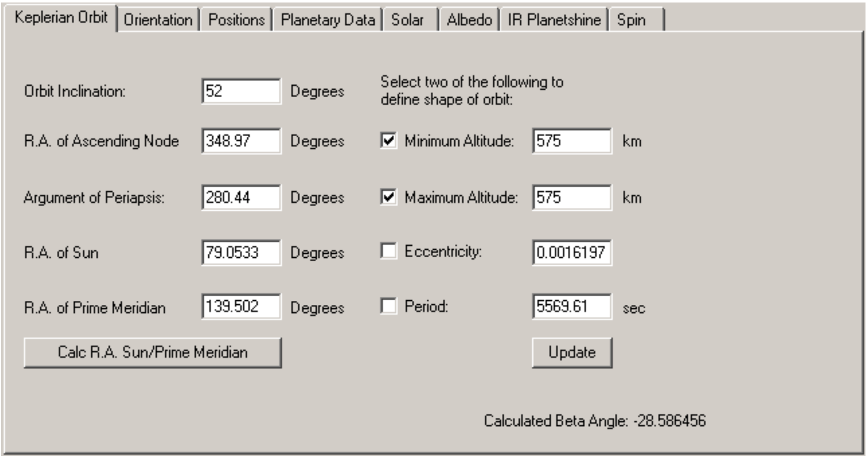


Name	Cond [W/m/K]	Dens [kg/m^3]	Cp [J/kg/K]
Aluminum	240	2694.4	900
Celsat	148	2330	700
Cooper	401	8940	384
HeatPipe	55	5317	319
PCB	ANISO	2200	603
Peltier	59.8	2	384
PeltierMicroPelt	5.97	2694.4	900

Figure 3.5 Thermophysical properties

3.3.4 Orbit parameters

The last part before starting the simulations is to define the orbit of the satellite. As it has been already defined in section 3.1 the only thing to do here is to copy all the parameters and write them as asked in the program.

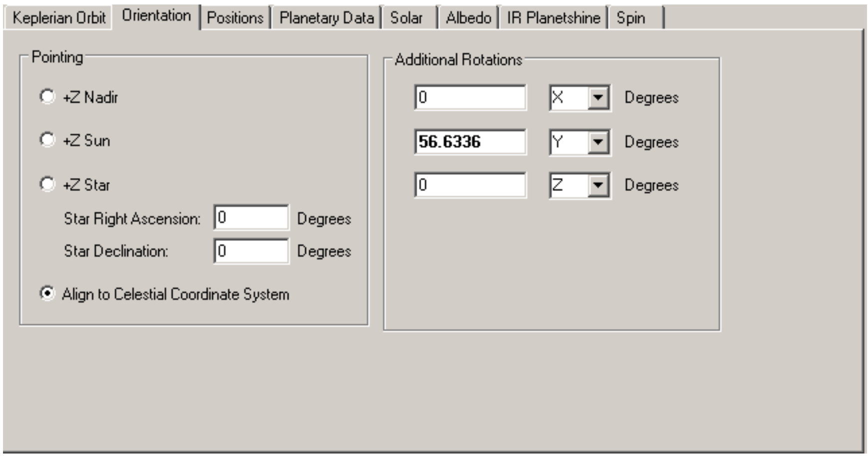


Orbit Inclination: 52 Degrees
R.A. of Ascending Node 348.97 Degrees
Argument of Periapsis: 280.44 Degrees
R.A. of Sun 79.0533 Degrees
R.A. of Prime Meridian 139.502 Degrees

Select two of the following to define shape of orbit:
☑ Minimum Altitude: 575 km
☑ Maximum Altitude: 575 km
☐ Eccentricity: 0.0016197
☐ Period: 5569.61 sec

Calculated Beta Angle: -28.586456

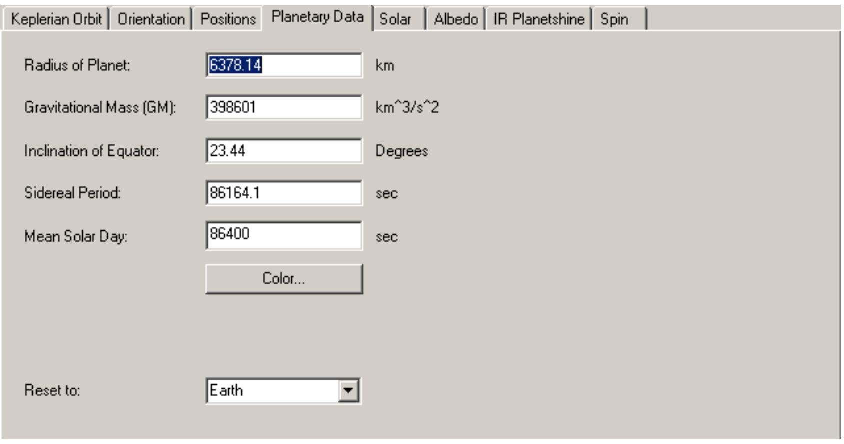
Figure 3.6 Orbit parameters 1



Pointing
○ +Z Nadir
○ +Z Sun
○ +Z Star
Star Right Ascension: 0 Degrees
Star Declination: 0 Degrees
● Align to Celestial Coordinate System

Additional Rotations
0 X Degrees
56.6336 Y Degrees
0 Z Degrees

Figure 3.7 Orbit parameters 2



Radius of Planet: 6378.14 km
Gravitational Mass (GM): 398601 km^3/s^2
Inclination of Equator: 23.44 Degrees
Sidereal Period: 86164.1 sec
Mean Solar Day: 86400 sec
Reset to: Earth

Figure 3.8 Orbit parameters 3

Finally, the orbit is shown in the thermal desktop so one can check that the parameters have been well written and that the orientation of the satellite is the desired.

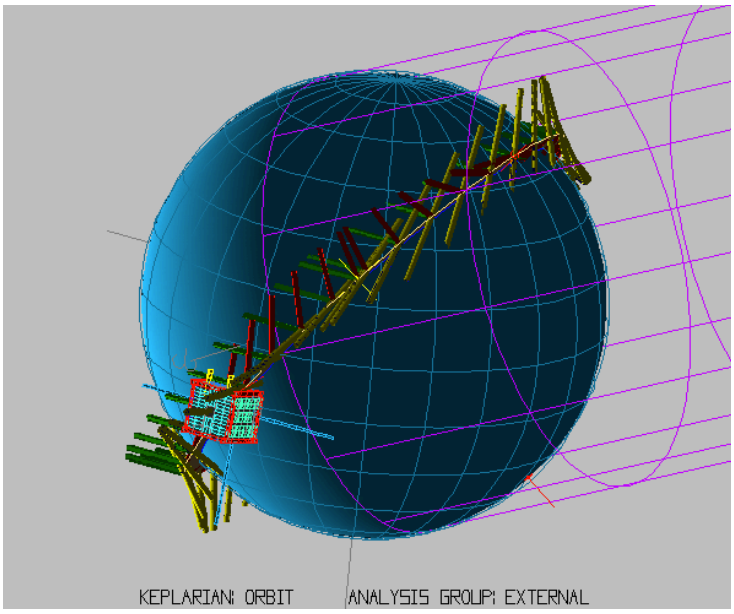


Figure 3.9 Thermal orbit

3.3.5 Simulations and results

Although these results give a clear idea about what to expect of the thermal environment, the inside part of the cube should be more detailed for better results. Moreover, the heat radiation inside the cube cannot be as detailed as wanted due to the lack of knowledge about this thermal field.

The results of the simulation can be presented in two different ways, having the temperatures overprinted in the satellite, and using the post-processing options, the temperature of a certain point can be shown in a graphic. The analysis has been separated in two parts, one regarding the outside of the cube and the other regarding the inside.

External analysis

This external analysis includes the structure of the satellite and all the faces, the solar panels and the antennas. The results of the simulation are shown in the next images, first with the model of the satellite and later with two graphics, one showing the point with the lowest temperature and another with the highest.

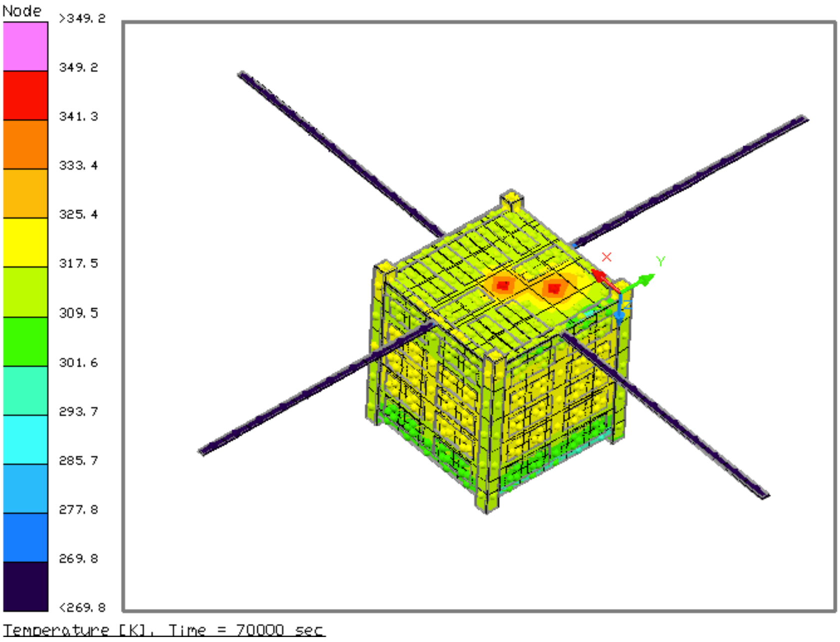


Figure 3.10 Satellite external analysis

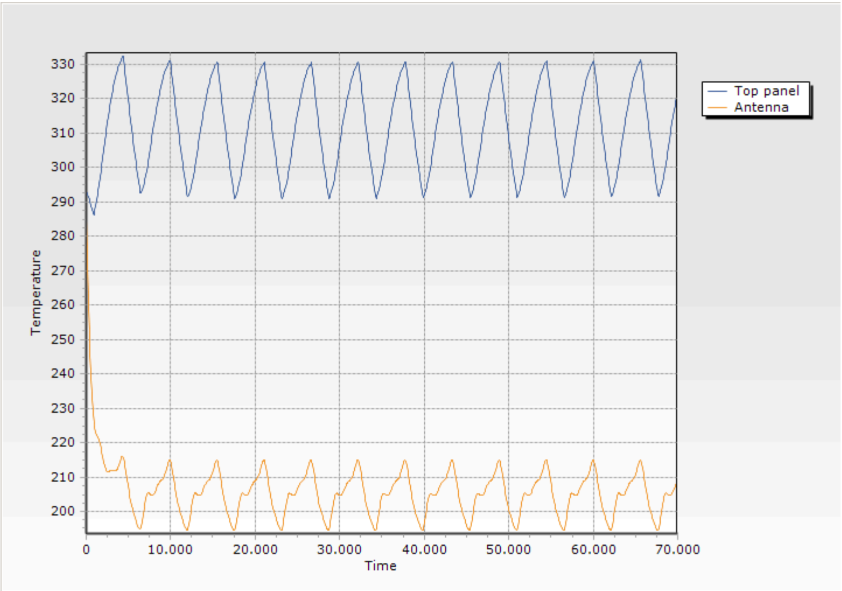


Figure 3.11 External analysis plot

After analysing the results, the temperature to expect in the outside part of the cube is between 20 and 50 Celsius degrees, what shouldn't be a problem for the satellite.

Internal analysis

This internal analysis should include the inside parts of the cube like the PCBs, and all the subsystems parts. In this case, it only includes the PCBs because a more detailed model couldn't be included.

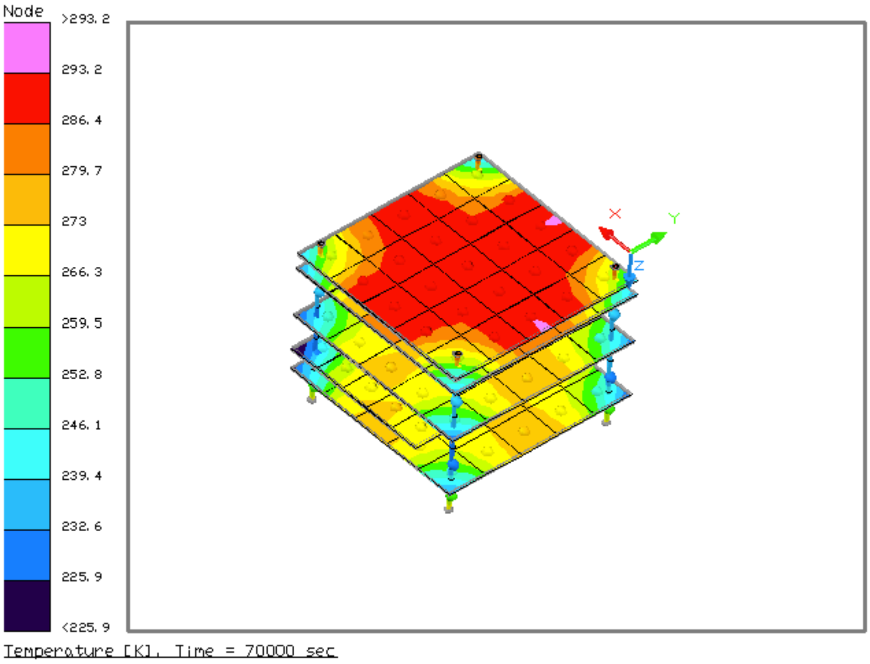


Figure 3.12 Satellite internal analysis

This simulation, despite not being very precise shows that the temperature in the inside of the satellite is lower than the in the outside as it doesn't have the Sun radiation. However, the real temperature will always be higher because when the subsystems are on, they dissipate heat.

The next image shows the temperature variation over time for the OBC and the EPS. It can be appreciated that the EPS temperature is a bit higher than in the OBC due to the kind of components that it has.

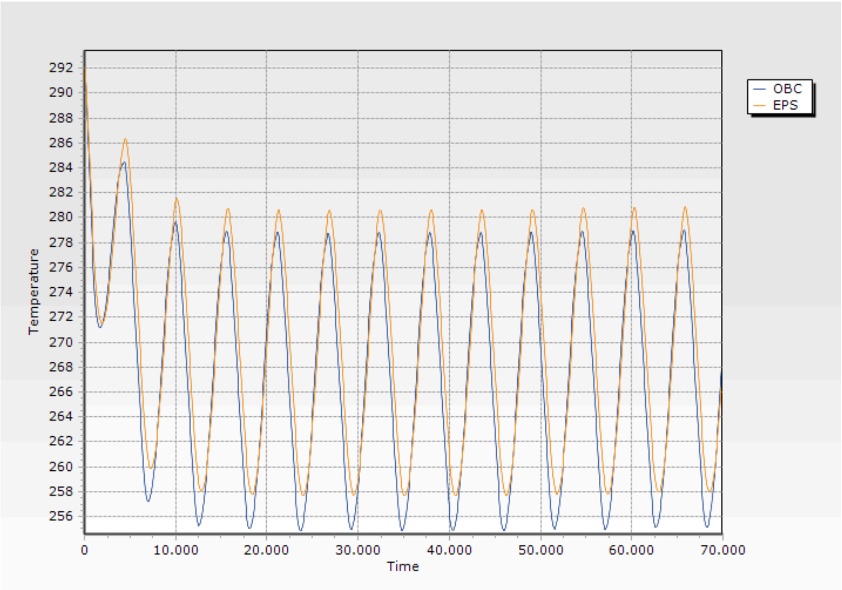


Figure 3.13 OBC and EPS temperature over time

Bibliography (Chapter 3)

[1] http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/orbit/ISS/SVPOST.html

[2] www.amstat.org

[3] *Final Design Review (FDR),* Projecte Avançat d'Enginyeria (PAE) – QP 12/13

[4] Help dialog from the STK SEET.

[5] **Lionel Jacques,** **T**hermal **D**esign of the **Oufti-1** nanosatellite, **U**niversity of **L**iège **A**pplied **S**ciences **F**aculty, **C**entre **S**patial de **L**iège

[6] **NASA,** Low earth orbit spacecraft charging design handbook, 2007

[7] PAE Cubesat, Thermal Analysis

[8] **A. Camps,** PAE PICOSATS presentation (Chapter 2, Orbits)

[9] **A. Camps,** PAE PICOSATS presentation (Chapter 3, Space Environment)

[10] **A. Camps,** PAE PICOSATS presentation (Chapter 4, Thermal Control)

[11] Kordon, M.; Baker, J.; Gilbert, J.; Hanks, D., "Space mission scenario development and performance analysis tool," *Aerospace Conference, 2005 IEEE* , vol., no., pp.4423,4432, 5-12 March 2005

[12] Elhady, A. M., "Design and analysis of a LEO micro-satellite thermal control including thermal contact conductance," *Aerospace Conference, 2010 IEEE* , vol., no., pp.1,11, 6-13 March 2010

[13] Ramirez, L.; Wang Jun; Liu Quojing, "Analytical time-dependent model for satellite thermal balance," *Microsystems Packaging Assembly and Circuits Technology Conference (IMPACT), 2010 5th International* , vol., no., pp.1,3, 20-22 Oct. 2010

[14] Stodden, D.Y.; Galasso, G.D., "Space system visualization and analysis using the Satellite Orbit Analysis Program (SOAP)," *Aerospace Applications Conference, 1995. Proceedings., 1995 IEEE* , vol., no.0, pp.369,387 vol.1, 4-11 Feb 1995

CHAPTER 4: INTEGRATION

When talking about system integration is not only taking the different subsystems and mixing them to build the hold system. This would be the last part of integrating a whole system, but before this, a lot of considerations must be done. Integration is about making changes in subsystems, taking care about compatibilities and taking decisions to achieve great results in the mission.

This chapter describes why and how the decisions have been taken during the whole process of building the system. Constrains that have been used as guidelines for the integration are the output results of the previous chapter. These outputs, lead us to three different main fields for the integration considering the physical and mechanical parameters, the energy balance, and the atmospherical conditions.

4.1 PHYSICAL INTEGRATION

As a system integrator, one of the most important things is to have a 3D model of the cube and of every piece of it. Having this, allows you to know how to fit all the parts and how to design every subsystem to not exceed cube dimensions. In the next lines the problems and limitations that where found while modelling the cube and trying to fit every subsystem are explained.

Considerations

1. ISIS STRUCTURE: Possibly the most limiting piece of all. While measuring every part to know where the screws to hold the PCB's were placed, it was realized that the hold structure was completely asymmetric and that it was less than 10cm height on the inside, what meant that it would be another aspect to consider when placing every board.

2. BUS DIMENSIONS: the length of the Portux's bus has been a big limitation when designing and placing the subsystems PCB's as it only fits between two of the structure screws.

3. BUS PINS: Due to the requirements of the payloads a pin plan was done and in some of the subsystems it became a big deal to choose which were the best pins to use, not due to electrical issues but to components placement. For example, the communications board needs a 5V supply but the pins that brings these 5V in the bus, was placed just under one of the antennas and it was inaccessible, so it had to be moved to another pin.

4. SOLAR PANELS DESIGN: as the solar panels were previously designed, the placement of the antennas and the camera was already defined.

5. DEPLOYABLES: the deployables such as the coils must be located at the lower level of the cube, as the solar panel had already been designed to have a hole at the lower part for this purpose. This apperture is not only used for the deployables, it is also used by the geiger counter to get the space radiation.

6. PAYLOADS BOARD: A 4x4 square space was given to every subsystem designer to develop their boards.

4.1.1 ISIS structure

The structure chosen for CubeCat-1 mission is ISIS 1U CubeSat structure. It has two killswitches in order to improve safety, and comply with ESA requierements. The main frame that holds the PCBs and the metal plates to enclose it and give it robustness against vibrations also composes it. Moreover, some more protection against the radiation is given with 1mm thick aluminium panels on the faces of the CubeSat.

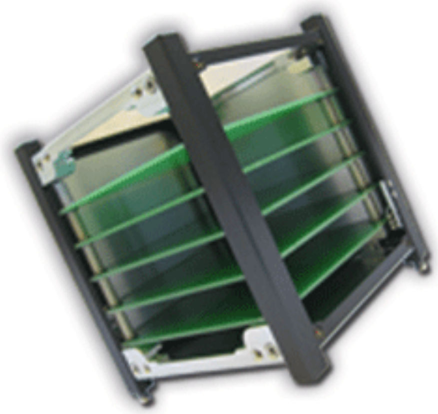


Figure 4.1 ISIS structure [3]

4.1.1.1 Asymmetry

When measuring and analysing this structure, it was realized that the columns holding the PCBs, where not placed in a symmetric way, what raised an important issue to consider when designing the subsystems' boards and adapting the Portux to fit into the cube.

Furthermore, not only the position of the columns wasn't as expected but also the inner part of the structure became a big challenge. The possible position of the PCB's is not the same if the satellite is considered side up or upside down.

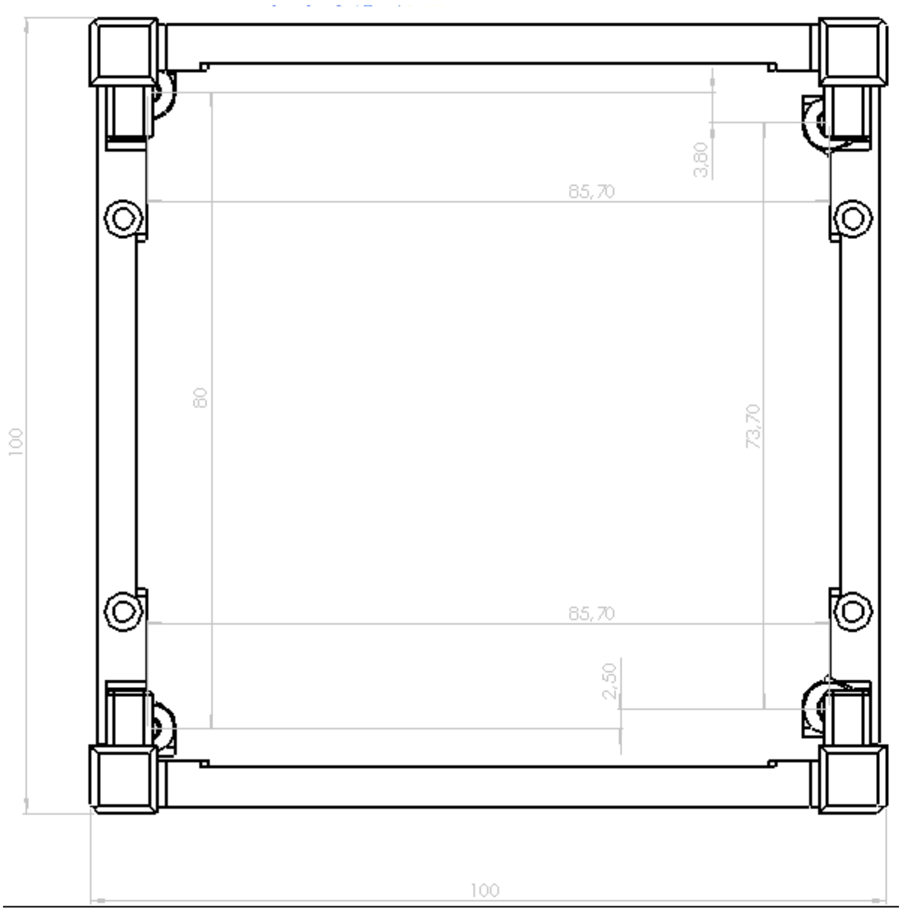


Figure 4.2 PCB's column's asymmetry

4.1.1.2 PCB layout design

The general design and shape of the PCBs is set by the ISIS structure, so it must satisfy the dimensions shown in the figure 19.

Moreover, the first floor not only has to fit but also has to hold the Portux board as it is shorter than cube dimensions and it only fits in two of the four columns. This Portux board, not only defines the structure of the first floor. It also determines the position of the boards respect to the cube sides because of the bus dimensions. As shown in the images below, the bus is 82mm long what makes that it can fit in only two sides of the structure.

The next image shows the PortuxG20 dimensions and as it is said before, the shorter size can be seen.

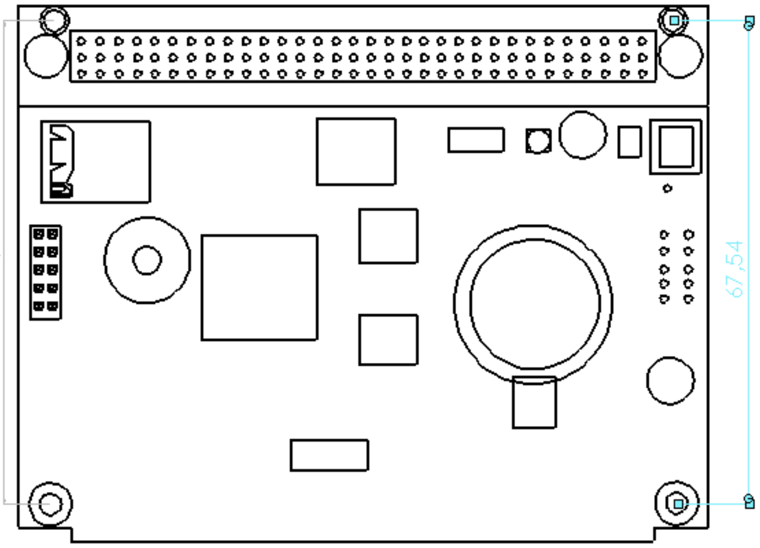


Figure 4.5 PortuxG20 layout

The following images show the dimensions of the PCBs and are used as templates for all the subsystems and can be a reference for future space missions using ISIS structure and PortuxG20 as OBC. The 96mm side size of the boards was considered in order to leave free space for the solar panels' power cables and the sensors on them.

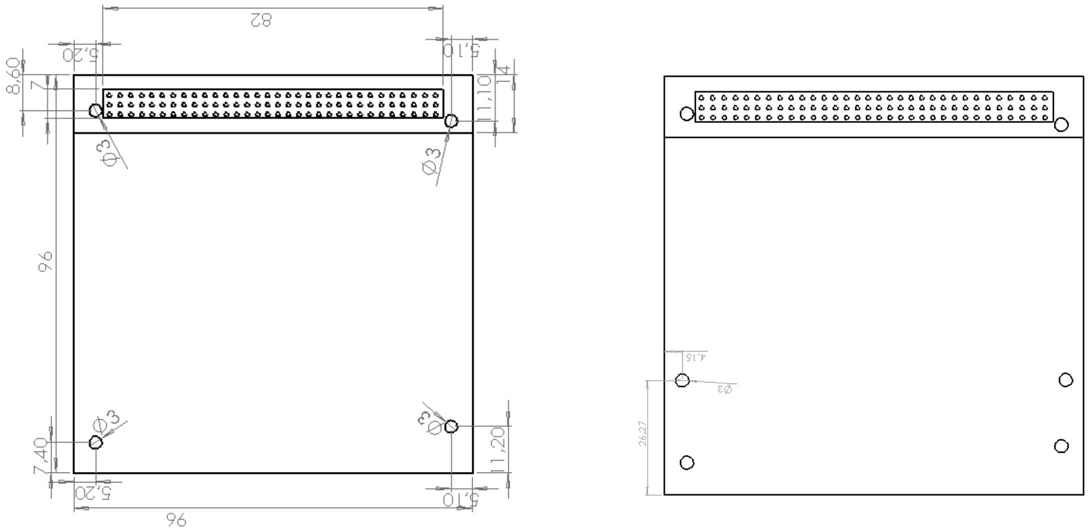


Figure 4.6 PCB layout

4.1.2 Floor distribution

Before starting building the whole cubesat in 3D, the floor distribution had to be done. The first and most important part is to know how many floors are needed to include all the subsystems and payloads, provided that the dimensions of the satellite are small and some of the systems have some components that need a lot of space.

The different subsystems to distribute were all the payloads and their deployables, the communications, the OBC and the EPS boards. Before this project started, the planification of the different floors had already been done but it had to be checked again as the devices were more advanced and some of them had new specifications.
The floor distribution was a distribution by type of subsystem. At that moment 5 different floors had been planed but didn't have a specific position to be placed. The floors were:

- Communications board: it needs a lot of space so a whole board for this system is requiered to hold everything necessary.
- OBC: the On-Board Computer is a device that is not designed and manufactured by the ^{3}CAT team. It is a computer bought and designed by Taskit. Due to its size, it needs a full floor for itself.
- EPS: this is the most complicated and delicated device in the satellite. If this board does not work properly, nothing can operate. The EPS contains the batteries, the sensors of the satellite and the power distribution for the whole system, so it also needs a full floor.
- PAYLOADS: these are all the experiments included in the mission. The objective here is to put all of them together in the same board so it is independent from the other subsystems boards and it is easier to supply power to them and get the information from every payload.
- DEPLOYABLES: due to the type of payloads in the satellite, it needs one board to include all the deployable systems. These systems are part of a payload but need to stay outside the cubesat during the mission. These parts are the coils from the WPT payload, and also, altough it is not a deployable, the geiger tube, which need a hole in the satellite's frame to get the space radiation.

Some of these subsystems have a fixed position due to its components. The communications board has to be on the top floor because of the antennas, and the deployables have to be on the bottom one. The solar panels and the satellite's shielding are already designed for these two parts. All of them have a hole on the upper part to allow the antennas go from the communications board to the outside and one of the side panels has a big aperture on its lower part thought for the deploy system of the WPT coils.

Once having the floors defined, it is time place them in the best way possible to minimize space and resources. Until this point, two of the boards have a fixed place, but still three of them need to be placed. So, the question "how to place them" must consider two different aspects, the data lines, and the power supply. Taking into account the data moving around the satellite, it can be seen that the subsystems need to communicate with the OBC and considering the power supply, every device has to be connected to the EPS. At this point, the most critical parameter will define which of the two devices, the on-board computer or the EPS, is going to be placed in the center of the boards. In the communication between the subsystems and the Portux, the difference in distance from having the OBC placed one or two centimeters higher or lower, does not affect much, but when considering power supply, the longer the cable is, the higher is the impedance that the current need to overpass.
Hence, the EPS is the subsystem placed in the center of the satellite, so it is equidistant from the upper and the lower boards.

Still two boards must be placed in the satellite, the payloads board and the OBC, and the possible placements are under the EPS or above it. This decision is easier to take than the one explained before. As it has been explained in the section before, the dimensions of the Portux do not fit the ISIS structure, so it has to be placed in a terrace on the top of another board. If it is placed above the EPS board, it means that the EPS loses part of its space due that it needs to have more holes to fix the OBC board, and as it has been explained before, it is a very sophisticated and complicated system and it needs a lot of space. So the placement of the on-board computer comes for itself; the deployables board only need to include the geiger device and the structures to deploy the coils, so it has a lot of free space.

Finally, the distribution of the 5 boards inside the satellite is the following, starting from number one in the lowest part and numer five in the highest:

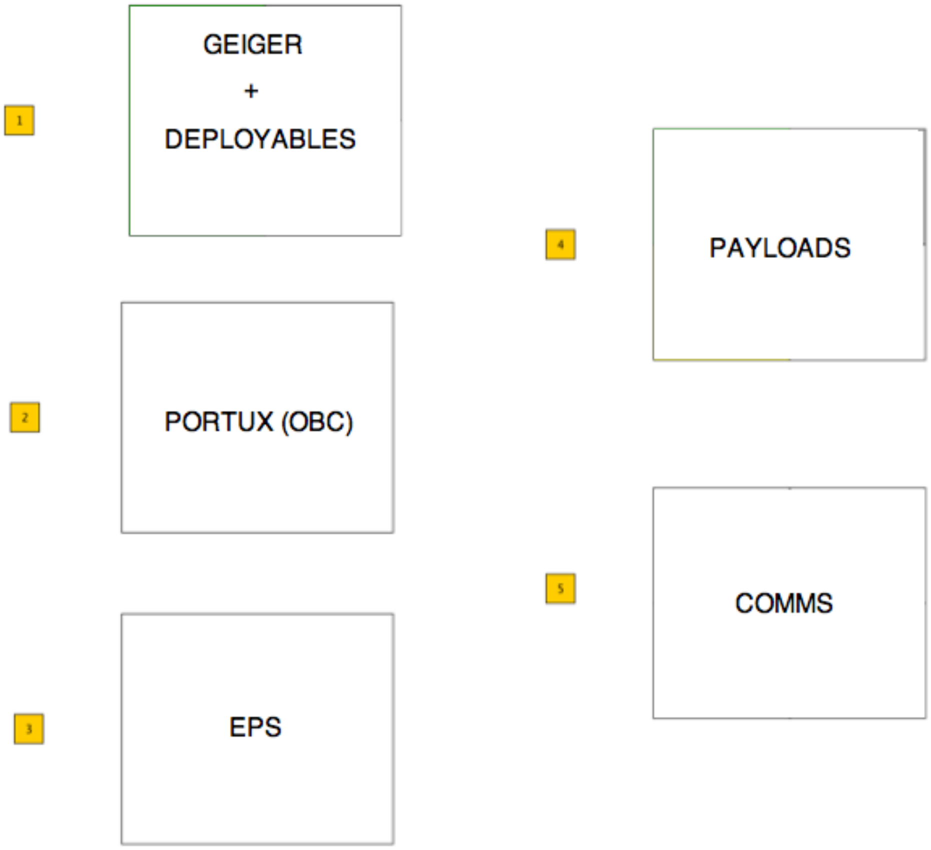


Figure 4.7 Floor plan v1

As the time passed, some modifications on the subsystems where made because some of the payloads' engineers asked for changing their location as it was easier for them to meet the specifications.

The most important change made was with the WPT and graphene payloads. These two subsystems were being designed by the same engineer and were supposed to be placed, as all the other payloads, on the fourth floor of the cube. The problem was that the WPT payload doesn't only have a circuit but also a deployable part, which should be placed on the first floor. At this point was when WPT engineer asked to move the circuit he had designed, both WPT and graphene payloads. The decision of moving this payloads to the first floor was made considering two different points; the first one, the space need for the PCB is 8x4 cm both sides, and it is the only place where there is enough space, and second but not less important, a cable is needed to communicate the WPT coils and their circuit, and it must have a certain length altough it can be adapted, and moreover, there is not much space inside the satellite to allocate cables.

After his request, the new floor distribution is the following:

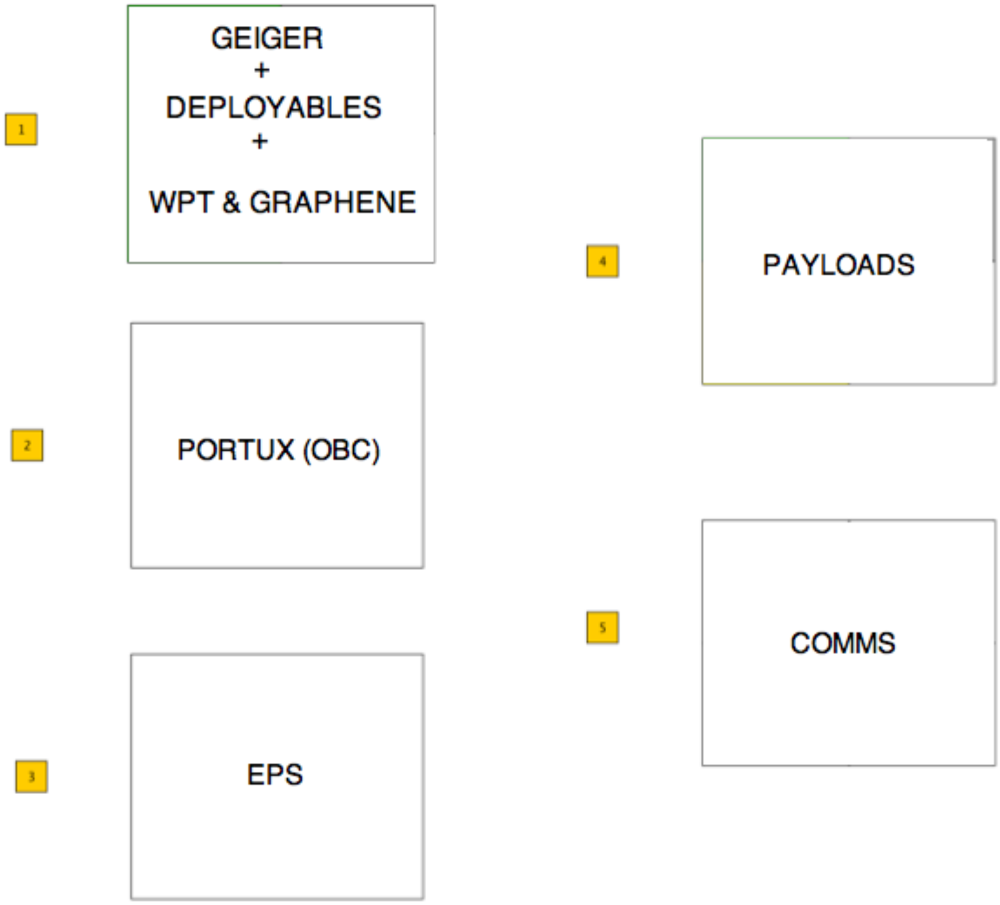


Figure 4.8 Floor plan

4.1.3 PCB design

It is important to do an equilibration of the satellite in order to avoid non-desirable rotations once the CubeSat is launched. The study of the center of mass defines the position of the heaviest parts of the satellite which limitates the PCBs design due to the heaviest components. Altough this is a very important issue to take into account, the distribution of each PCB is not only limited by the weight of the components but also for their size, as the biggest ones cannot be placed one on top of the other because if this is done, there is not enough space to put them all.

4.1.3.1 Center of mass determination

The aim of this study is to determine the center of mass of the CubeSat to avoid destabilization once it is in the orbit. There are two ways to do the center of mass determination. One of them is to do it empirically, but to do it this way, one needs to have all the components placed into the satellite, and the only thing that can be done if the center of mass is not placed in the right place, is to add extra weight to compensate it. The second, and the one described below, is using software that allows you to calculate where the center of mass will be one the satellite is assembled.

It is very difficult to have a perfect simulation as it would need a lot of time to model all the parts and define their weight and center of mass and later, put all of them toghether to get the satellite's center of mass. However, usign software, a first aproximation can be done only defining the biggest pieces and the results can be used to determine the placement of them and the design of the PCBs.

As a model of the satellite had been started using SolidWorks and it allows including weight characteristics, it was decided to do it using this software. SolidWorks is 3D design software that has a lot of functionalities such as 2D and 3D modeling, sketching, assigning weight properties… One interesting issue of SolidWorks is the possibility of doing assemblies, for example: the Portux is an assembly of various components (USB, Ethernet…). With this software, each component can be designed separately, and after that do an assembly with all the components. The first model included the CubeSat skeleton, Solar cells, antennas, Camera and Portux. These components were fully designed but not assigned their weight properties, so the first task was to assign them.

4.1.3.1.1 Components characterization

This software was unknown initially, so the first step was to do some tests with "camera.SLDPRT" properties and finally it was totally defined in size and weight.

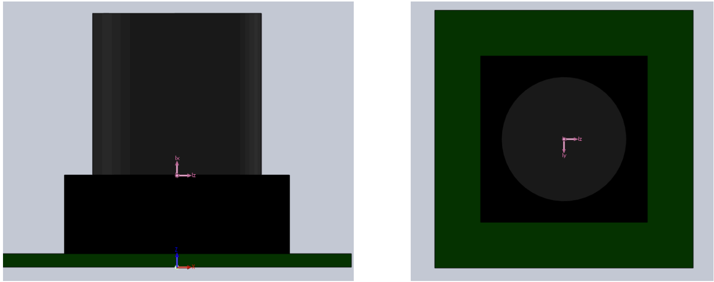


Figure 4.9. Camera center of mass

Once the camera was fully characterized, the next step was to work in the Portux, which is an assembly composed by different pieces. It has a PCB at the bottom and on the PCB there are two USB ports, an USB ASP port, a VGA port and an Ethernet port. They have different size and weight and, as it was measured at the laboratory, but these conenctors are going to be removed from the board for the mission, so their weight was not considered. Without the connectors, the whole Portux weights 22 g and its center of mass can be seen in the next image.

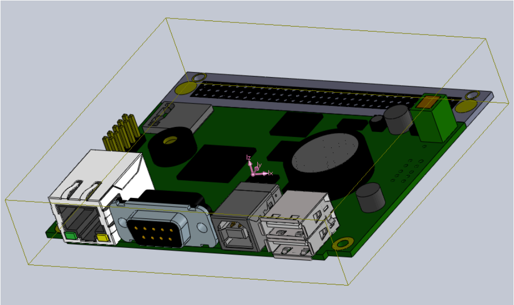


Figure 4.10. Portux center of mass

To continue with this work, as not all the components were available to get their weight specification, the heavier ones were modeled and their characteristics are contained in the next table.

Subsystem	Type	Weight [g]	Size [mm x mm]	Height [mm]
ADCS	Coil	13	39.9 x 12	12
ADCS	Magnet			
EPS	Batteries (Duracell DRF60) x2	27 x2	35.3 x 54.6	19.5
EPS	Solar cells x6	276	98 x 82.6	3.6
OBC	PortuxG20	22	97.5 x 75.2	12.3
PAYLOADS	Camera	11	32.2 x 32.2	26
PAYLOADS	Geiger	53	100 x 44.4	30
SI	Skeleton	91	100 x 100	100
SI	Floor	12	100 x 100	

Table 4.1. Heaviest component characteristics

The measures that are shaded in red are the ones that may vary because the components are provisional or are in an improvement process. As it can be seen, the batteries are the heaviest component appart from the solar panels and the structure. The models below for the batteries and the ADCS coil show their characterization and their center of mass.

For the batteries, the center off mass is the same as the geometric center.

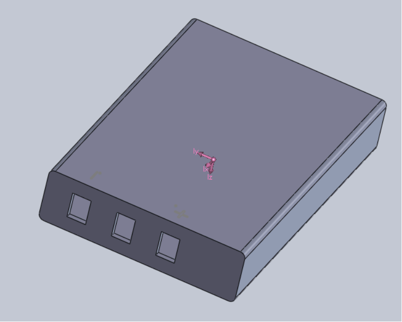


Figure 4.11. Battery model and center of mass

For the coil modeling two pieces were used: the core and the coiling. The file "ADCSCoil.SLDASM" is an assembly of these two pieces and represents the coil of the Attitude Determination and Control System (ADCS). As it is symmetric, its center of mass coincides also with the geometrical center.

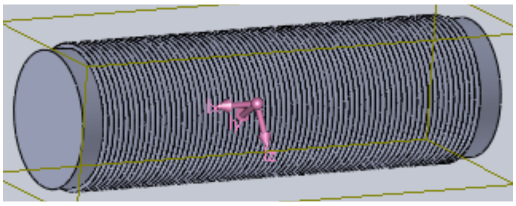


Figure 4.12. Coil model and center off mass

The magnet was not available to measure so it was supposed that its size and weight would be similar to the coil's ones, so a component was done with these size and weight, but with a rectangular form.

The Geiger counter was also unavailable yet so it was modeled as a box with its expected size and characterized by a weight of a first version of it.

The structure of the CubeSat was already modeled and its size was fully characterized, so it was only needed to put the correct weight of each component of the structure (Skeleton, Solar panels and floors). It is almost symmetric, so the center of mass of this assembly coincides with the geometric center, and as it represents the heavier part (near 400 g), there will be a good margin to do the distribution without displacing much the center of mass.

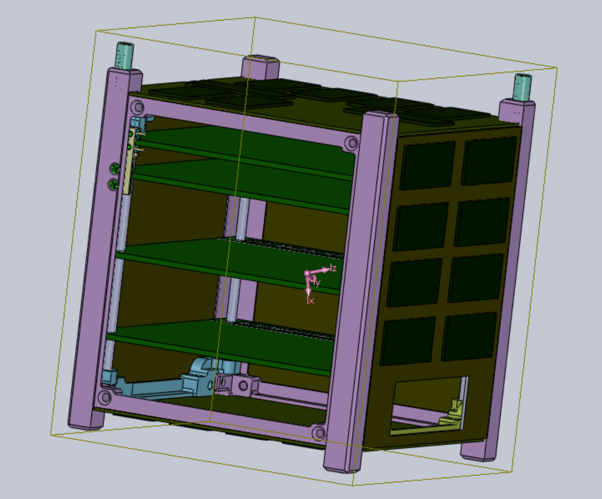


Figure 4.13. Model of the cubesat structure and its center off mass (front panel hidden to show the inside)

4.1.3.1.2 Center of mass

Based on the properties of the components, Solidworks calculates the center o mass, and shows it in the 3D model. What is needed to know is the distance from this point to the geometric center. To do that, the coordinates of these two points have to be known, construct a vector, and calculate its magnitude.

However, the information at this point is not enough to determine the center of mass because only the weight and the center of the heaviest components is defined but not their place in each PCB. Having the floor distributed and explained in the section before, the placement of the components needs to be defined. The reasons why these components are placed in a point and not in another one are explained in the next section, 'PCB design'. This PCB design section does not explain how the center of mass is considered to place the components but it has always been an item to consider; the heaviest components need to be well distributed to equilibrate the satellite. Anyway, not only the weight is a point to consider but, as commented before, the dimensions of the components are also an important characteristic that may force some component's placement.

After updating the layout of each floor, the new center of mass can be calculated with SolidWorks. The result is really good because, as we can see in figure 4.14, there are two columns of heavy components and they are compensated by their weights:

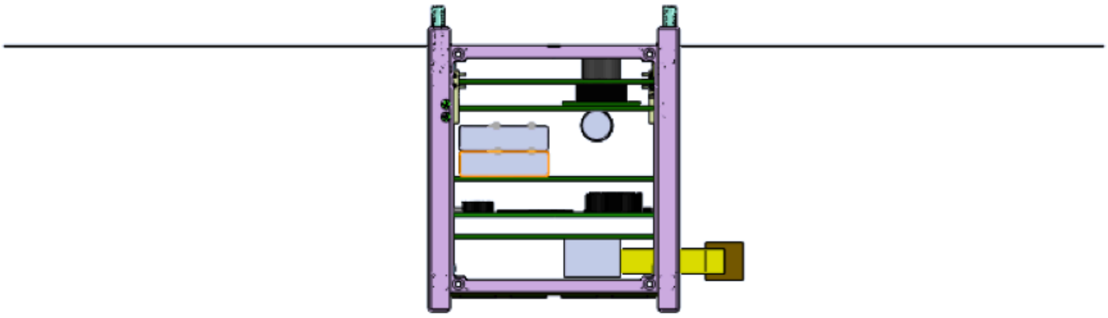


Figure 4.14 Cubesat distribution

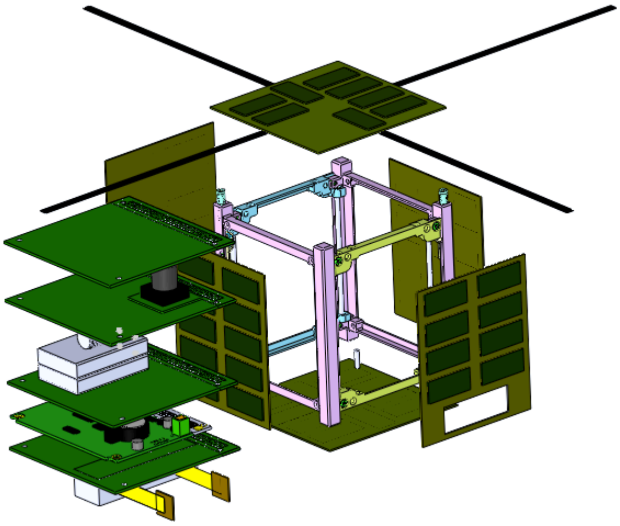


Figure 4.15. Cubesat explosioned view

The heaviest component on the CubeSat are the batteries, and they are placed opposite from the camera, the coil and the Geiger counter, which are the other heaviest components.

In the vertical direction, the batteries are allocated above the PCB and this has been moved down, in order to do not be detrimental for the center of mass, and if this displacement is done in a good way, the results may be really favorable. These results are taken from the SolidWorks and they are the geometric center and the center of mass. After obtaining these values, the distance between them can be calculated and therefore, know if the center of mass needs to be moved ading some extra weight or not.

$$Center\ (geometric) = \left(-7'\ 31,2'\ 21,4\right)\ mm$$

$$Center\ of\ mass = \left(-7'\ 11,2'\ 12,5'\ 38\right)\ mm$$

$$\overrightarrow{Distance} = \left(0'\ 2,0'\ 09,1'\ 38\right)\ mm$$

In conclusion, it can be observed that there is an important part of the weight that is well equilibrated (which is the structure), and that proportionates a bit of freedom to move some items without displacing the center of mass so much.

However, it is important to keep in mind that the components included in this model are only the ones that have a weight greater than 10 g. The complexity of the CubeSat makes impossible to introduce all components in the model, because there are some pieces that cannot be included, such as wires, screws…

That's why this study is not 100% reliable, and it will be necessary to do an empirical study and equilibration, when all the components are ready to be assembled. At this point it is very important to remember that the maximum distance between the geometric center and the center of mass can be at maximum 1 cm and that the satellite can include 300g of extra weight from 1 Kg to compensate this situation if needed.

4.1.3.2 PCB design

Once the 3D model was done, it became much more easier to design the PCBs of every floor given the placement of the biggest parts.
Apart from this, one decision had been done before having the model, and it was to have a 4x4cm square for each payload in their floor so all of them could fit.

The next lines explain the considerations taken when designing every PCB and all the problems that had to be overcomed.

Comms board

This board had only one thing to consider, the hole for the camera. The top solar panel already had the hole done and due to the position of the screws to hold this panel in the structure, the camera had only one possible placement.
The image below shows the communications boards with the camera's hole.

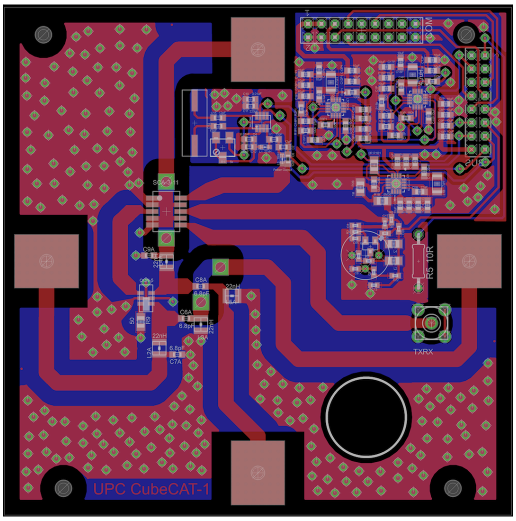


Figure 4.16 Comms board

EPS and payloads boards

This board has one of the biggest items of the satellite, the batteries. The best way to allocate the batteries would be to put one on each side of the PCB, but it is not possible as they need a heater to maintain their temperature and it must be placed between them. Another point where the batteries become important is when considering the board on the top as the batteries are placed on the top face of the EPS. The board on the top of the EPS is the payloads floor, which contains the attitude coils placed on its bottom face.

The first idea after encountering this problem, is to move the attitude coil to the top side, but there is where the camera is placed, and it has to be as near as possible to the communications board to reach the outer part of the satellite. Then, the thing is to place correctly both the batteries and the coil. After a lot of iterations, the distribution of these two boards is the one showed in the image below.

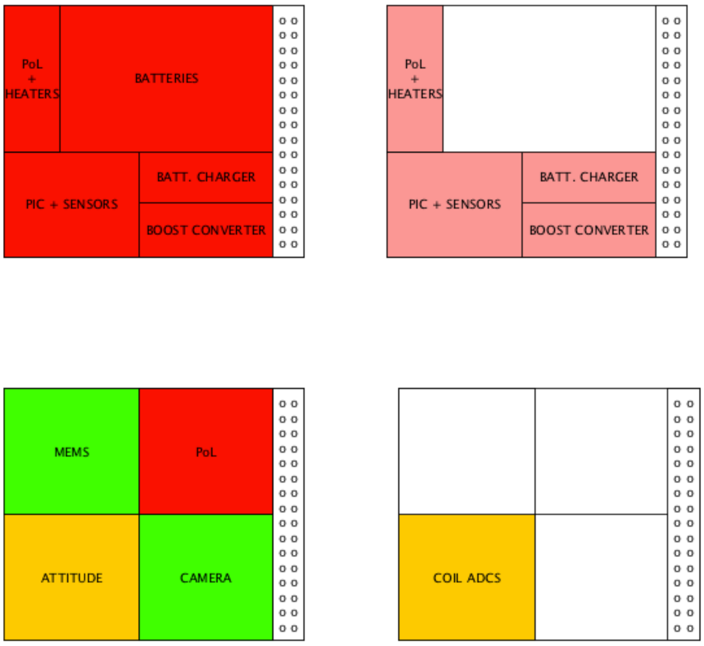


Figure 4.17 PCBs' distribution

Deployables board

When the mission started, this board was supposed to include only the deployables and the Geiger tube, but as explained in the previous section it finally includes these devices and also the peltier and the WPT and graphene payload.

The things to take into account here are, first of all the deployment system (still to be finished), and the position of the Geiger tube. The tube must receive radiation from space, so is has to be near the solar panel's hole. Another decision taken related to this board is to place it with all the devices on the bottom face. This is done due to the Isis structure; when placing the bottom board, between this and the structure, a screw must be placed, increasing a lot the separation between the board and the rail of the structure. If the components are placed on the bottom side, this free space given by the separation is used.

Despite having the components placed on the bottom side, the distribution of this first floor is the one showed in the picture.

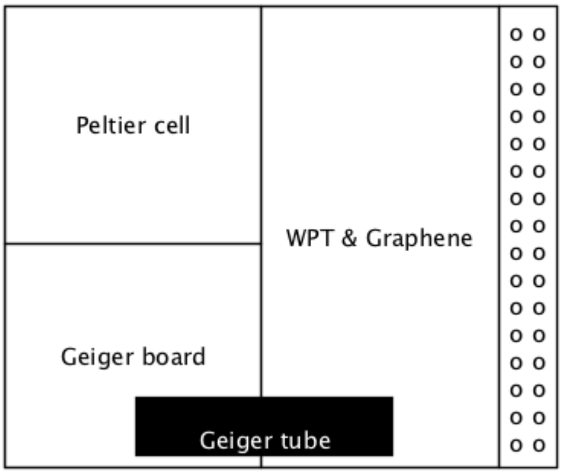


Figure 4.18 First floor distribution

4.2 BUS DISTRIBUTION

Data lines and power supplies must be placed into the cube to communicate payloads, subsystems and the OBC. The on-board computer used, Taskit Portux G20 includes a DIN 41612 bus containing the following pins [8]:

- 1 Synchronous Serial Controller (SSC, I2S)
- 1 Serial Peripheral Interface (SPI)
- 1 Two Wire Interface (TWI, I2C)
- 1 MultiMedia Card Interfaces
- 5 USARTS
- Digital Ports - up to 64 available
- 4 Programmable Clocks
- 4-channel 10-bit ADC
- 16-Bit parallel CPU-Bus

It was decided to use this bus to have all the data connections and also the supplies. As some of the various functions are realized by multiplexing connector pins and therefore not all functions may be used at the same time, the pins are assigned given the pin plan of the portux found in [8] and shown in the appendix C of this document.

The next section includes all the subsystems needs related to connections.

4.2.1 Subsystems requirements

Before planning the bus for all the connections it is necessary to know all the connections needed for each subsystem. These connections involve both data and power lines. First of all, and before defining the type of connection needed, it is important to know which are the connections, which systems needs to talk to another and how the supply arrives to each part.

The next two diagrams show all the connections, one for data and the other one for the supply.

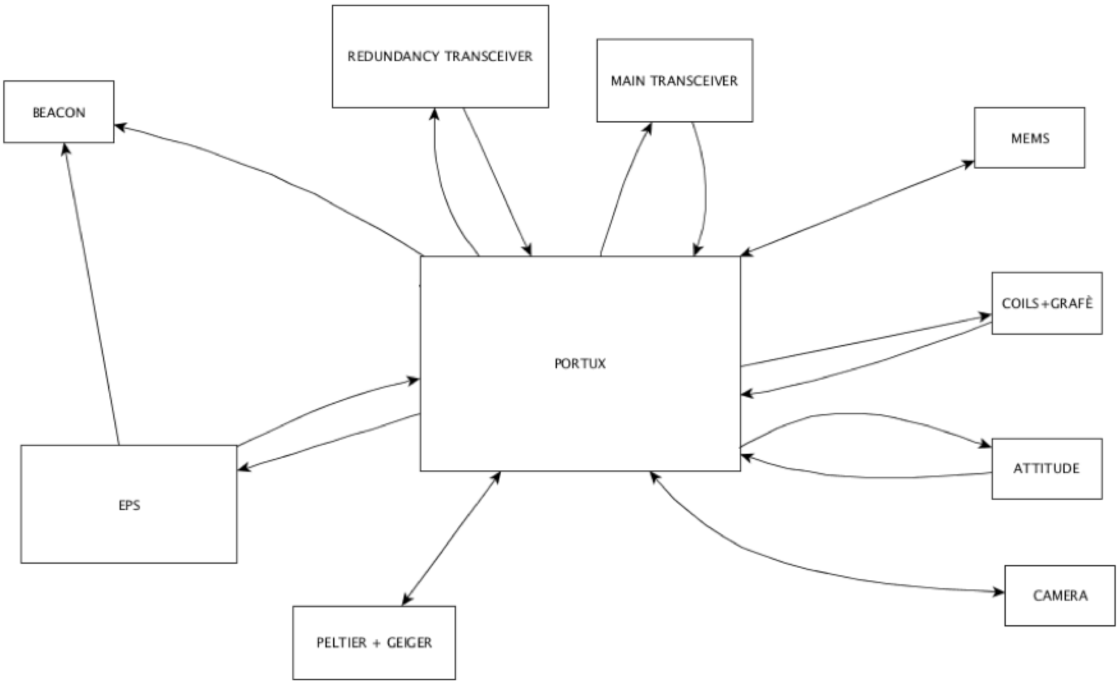


Diagram 4.1. Data connections

As can be seen here, the data connections go from each subsystem to the Portux and there are connections in both ways, as the Portux needs to receive information from the payloads but they also need to get commands from the OBC.

The electrical connections for the supply go from the EPS to the subsystems and the communications board also has a connection with the peltier which supplies one of the beacons.

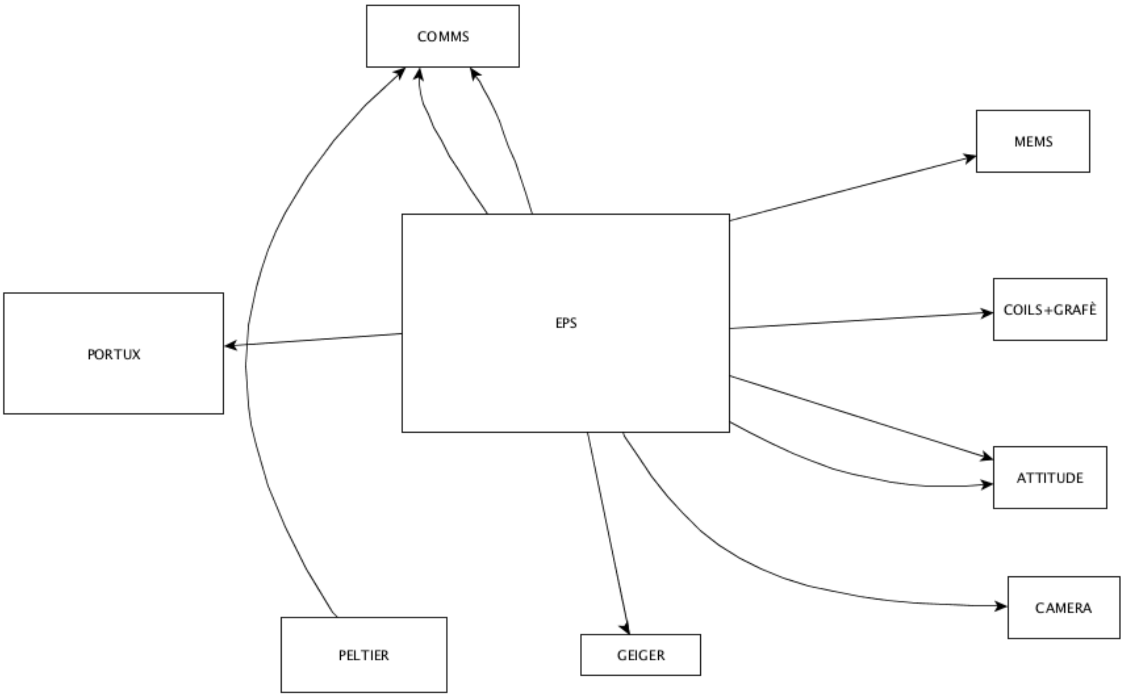


Diagram 4.2. Electrical connections

All the connections, both electrical and data of each subsystem are explained in the next lines. The code after each connection indicates the position in the bus and this distribution is explained in the next section.

EPS

This system is the core for the electrical connections, what means that all the other subsystems are connected here to get their supply. It needs the connections for the power supplies and also for the point of load (POL). The connections are:

- 2 lines for power supply (B8, C8)
- 1 ADC converter (A25)
- 8 POL to enable the subsystems (B2, B6, B12, B17, B23, B26, B27, B29)
- 1 Tx, Rx line to communicate with the Portux (A10, A11)
- 1 line for the antenna deployment (A1)
- 1 digital line for the beacon (A6)

COMMS

The communication board needs to receive the information from the Portux to transmit it to the ground station and this is done through SPI. It also needs the power supply and the connections for the peltier beacon.

- 2 POL enable for 3.3V and 5V (B2, B6)
- 1 SPI connection with the Portux (A3, A4, B4, B5)
- 1 digital input for the EPS-Beacon communication (A6)
- 1 digital connection with the peltier (A7)
- Gnd and Vcc for the peltier beacon (A8, B7)
- 1 line for the antenna deployment (A1)

WPT and GRAPHENE

These two systems, that are designed to be contained in one only board, only need one enable pin as it can convert the voltage from the EPS to the values needed and an SPI to the Portux.

- 1 SPI with the Portux (A4, A14, B4, B5)
- 1 POL enable (B12)
- 1 digital input pin (A12)

ATTITUDE

This system only needs Tx and Rx lines and the power supplies.

- 1 Tx, Rx line with the Portux (A2, B3)
- 2 POL enables for 3.3V and 5V (B26, B27)

CAMERA

This payload also needs a line for Tx and Rx and only needs one POL of 5V.

- 1 Tx, Rx line with the Portux (A15, B16)
- 1 POL enable for 5V (B17)

MEMS

Like the two subsystems above, the MEMS need one POL and a Tx, Rx line. The MEMS payload needs voltages of 5, 3.3 and 1.8 V, but it is prepared to generate them by itself using the input from the EPS.

- 1 Tx, Rx line with the Portux (A29, B30)
- 1 POL enable for Vcc (B29)

GEIGER COUNTER

This payload only needs a supply and a counter line to the Portux.

- 1 POL enable for 5V (B24)
- 1 counter pin (B23)

4.2.2 Pin assignment

Once knowing all the pins needed for each device, the pin plan was done. Not many considerations were done while planning, only the type of connection needed, and for the comms board, the position of the antennas. Comms board can only use the sides of the bus as it has the antennas placed in its center, so the first thing to do with the plan was to appoint communication's pins.

Once knowing what kind of connection need each subsystem, the bus pins can be assigned. The motto is to have the pins of each subsystem as near as possible, taking into account that this is not always possible, for example for the SPI ports. This type of connection needs 4 different pins and three of them are the same ones for all the devices.

The priority here, as said before, is to place the connections of the communications board, as the antennas do not allow having the full bus in the board. When discussing where to place them, together with Sumit Karki, it was decided to have them in the left side of the board, at the first pins. Then, the other connections are placed considering the type of connection of each subsystem and also placing them as separated as possible of each other to avoid interferences.

After having everything placed, a last decision was taken to avoid power supply problems. This was to have two different pins for the power supply in order to stand the current provided by the EPS.

The next diagram shows the pin distribution and the detailed pin plan document is available in appendix D.

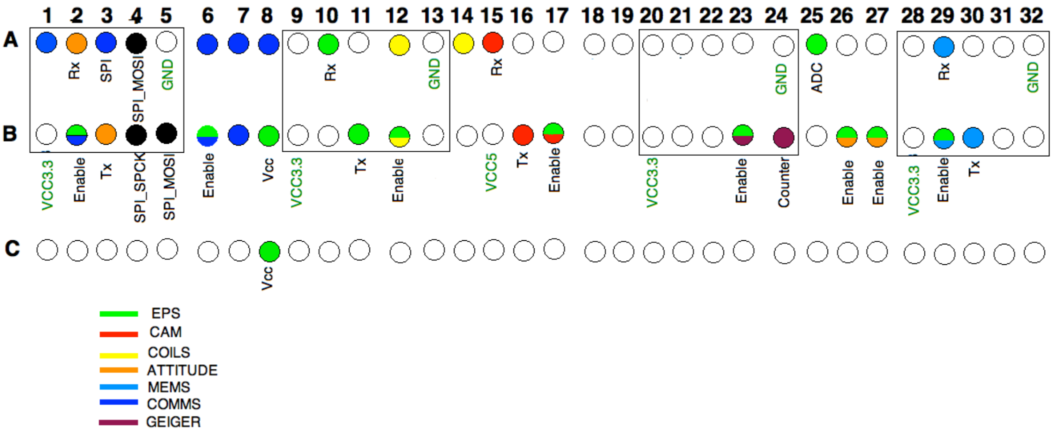


Figure 4.19 Pin distribution

4.3 AVOIDING HARMFUL ENVIRONMENTAL CONDITIONS

4.3.1 Batteries' heater

Due to the results extracted from the thermal analysis, it was decided to put a heater for the batteries to avoid them to get frozen. This heater is placed between the two batteries and it is controlled by the EPS, which has a temperature sensor and switches on and off the heater.

4.3.2 SD card

In our cubesat, some of the devices like the OBC, the graphene and WPT payloads use a micro SD card. Not all kinds of SD card can survive space's temperature and radiation, so we needed to find a card to withstand the conditions of the mission. After searching a lot, it was found the micro SD that will be in our mission. [1]

Figure 4.20 microSD card

*SanDisk Extreme Pro microSDHC UHS-I Cards are designed to handle whatever life throws at them. This card is shockproof**, X-ray proof**, and waterproof**. You can play your apps, capture memories and access data in almost any climate--the card operates in temperatures ranging from -13 to 185 degrees Fahrenheit (-25 to 85 Celsius degrees). Take your phone or* *tablet to the snow, to the pool or to the desert. Your card will survive, even if your device doesn't.*

4.4 MISSION VIABILITY

This section is the final and most important one in this document as it determines the viability of the mission considering both energy and data of each subsystem. First of all, two different simulators are described here, the energy simulator and the data simulator, but to be able to decide about the viability of the satellite, both need to be combined and simulate them together in order to be sure that the constrains used in each of them are compatible.

4.4.1 Energy management

Cubesats have small physical size, which restricts the size of the solar panels and thus the available power budget and stored energy reserves. This makes the energy balance a very important issue to take care about. The on-board computer is going to have a software module that will take care about the state-of-charge (SOC) of the satellite and it will decide whether to run or not a task.

However, before implementing this software part and upload it to the satellite's computer, it was decided to implement an energy simulator to have an estimation about the energy behaviour of the cubesat.

4.4.1.1 Energy manager simulator

As explained before, the energy control in a cubesat is very important due to its small capacity to get and store energy. Knowing this, the idea of having an energy simulator arised. This simulator will have two main functions, one, as tells the name, simulate the behaviour of the cubesat in terms of energy, and the second, be a reference for the energy manager included in the OBC. The first idea of the simulator was to have the income and stored energy profile and run a random number of simulated payloads and to have an output variable containing the answer to know if that simulation had worked without running out of energy or not.

Later on, the next approach and the one that became the final was to have a graphic with the state-of-charge profile not only of the batteries but also of the whole system, which implies also considering the input energy when the batteries are full charged. State of charge (SOC) is a term that describes the percentage of full charge remaining in the batteries of electric or hybrid vehicles. This concept expresses how "full" the battery is as a percentage value with 100% being fully charged and 0% being empty or flat.

Although this is totally true, in this case, when referring to SOC means the available energy in that time instant and it is not measured as a percentage but as Watt·h. Empirically, the way to calculate the state-of-charge of a battery is to integrate the power received through time taking into account the efficiency and the batteries' charge and discharge profiles.

It was decided to implement this software using Matlab, as it is a good tool for the data treatment and for having detailed and analysable graphics for post processing. The next sections explain how the software is implemented, its data input and output and how to work with it.

4.4.1.1.1 Input and output data

The data needed for the simulator involves a lot of terms, going from the batteries capacity to each subsystem's energy profile, energy input, efficiencies and system deterioration.

Subsystems energy profile

This information comes from de Power Budget part (section 3.2.2) which contains information about how much energy does every subsystem consume and how long it needs to be on in each execution. These parameters correspond to the energy output.

Solar panels characterization

The characterization of the solar panels has been taken from a final degree project called 'Desarrollo, integración y caracterización de los paneles solares y el sensor óptico' [4]. This information refers to solar panels efficiency. For spectrolab cells, which cover three of the side faces of the cube and the top and bottom ones, the efficiency is between 26% and 29% and for the Cellsat cells, it is about 13%.

This information was initially placed as a parameter in the simulator, but after integration the Solidworks model to the STK software to obtain the input energy computing all the orbit parameters, the efficiency was included in the physical model so the computed energy input already includes this parameter.

EPS efficiency

Despite the energy goes to the solar panels, the device that receives, stores and distributes the energy is the EPS system. When taking the energy from the solar panels, it uses some kind of convertors, which are not 100% efficient, and it has to be taken into account. Their efficiency is around 85-90%.

Batteries properties

Another important thing is the way that the energy is stored, and it depends on the type of batteries used. In this case, the batteries used are the Duracell DRF60 [5]. The cubesat has two of these batteries which each of them is able to provide 1230 mAh and 3.7 V, what means 4.55 w·h. Also the degradation of the batteries is considered in the algorithm, as they will degrade through time.

Energy input

This is for sure the most relevant information in the energy simulator along with the subsystems energy consumption. The incoming energy of the satellite will depend on the orbit and the position of the satellite. The solar panel power received in the solar panels is computed, as referenced before, using STK software as it is an orbit simulator and can provide information about the sun radiation and therefore, the power obtained. More detailed information about how to obtain the input energy using STK can be found in [7].

The way to make this information accessible from the simulator is to convert the STK output file to a csv. file and read it using the code for this purpose (section 4.3.1.2).

An example of the csv file is shown in the next image.

```
Time (UTCG),"Power (W)","Solar Intensity"
 16 Oct 2013 00:21:00.000,0.071,1.000000
 16 Oct 2013 00:22:00.000,0.171,1.000000
 16 Oct 2013 00:23:00.000,0.276,1.000000
 16 Oct 2013 00:24:00.000,0.369,1.000000
 16 Oct 2013 00:25:00.000,0.464,1.000000
 16 Oct 2013 00:26:00.000,0.554,1.000000
 16 Oct 2013 00:27:00.000,0.648,1.000000
 16 Oct 2013 00:28:00.000,0.733,1.000000
 16 Oct 2013 00:29:00.000,0.821,1.000000
 16 Oct 2013 00:30:00.000,0.893,1.000000
 16 Oct 2013 00:31:00.000,0.991,1.000000
```

4.4.1.1.2 Matlab code and execution

The full code can be seen at appendix A, but in this section it is explained the most important part of the code that is how to calculate the SOC.

```
for i=1:T-tini
     if i==1
         SOC(i)=e_inici+(pin_wat(i)-Pout_wat(i))./3600;
     else
         SOC(i)=SOC(i-1)+(pin_wat(i)-Pout_wat(i))./3600;
     end

     if SOC(i)>cmax_degradacio,

          if Pout_wat(i)~=consum_portux/1000,
             %%%instant power%%%
             SOC(i)=SOC(i-1)+pin_wat(i)-Pout_wat(i);
             if SOC(i)<cmax_degradacio,
                 SOC(i)=cmax_degradacio-(pin_wat(i)-Pout_wat(i))./3600;
             end
             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          else
            SOC(i)=cmax_degradacio+pin_wat(i);
          end
     end
 end
```

This part of the code calculates the energy state of the satellite as an incremental addition, what is equivalent to an integration operation. The first lines are only to differentiate if it is the first iteration of the algorithm or not, as if it is the first one, the initial energy has to be considered. The next lines are written to treat the case when the batteries are fully charged. This is the point where the difference between energy and instantaneous power has to be considered. When there is energy received from the sun and the batteries are charged, the devices can be directly fed with the energy coming from the solar panels without passing through the batteries. When considering this case, the energy level only depends of the difference between incoming and outgoing energy, but sometimes, the energy required for some of the subsystems is higher than the one coming directly from the sun and therefore, the batteries state has to be also considered.

A tutorial on how to run the simulator is placed in the appendix B.

4.4.1.1.3 Simulations

The first simulation shown below illustrates how the energy coming from the sun is stored in the batteries until they are fully charged and after this happens, how the energy is received to feed directly the subsystems.

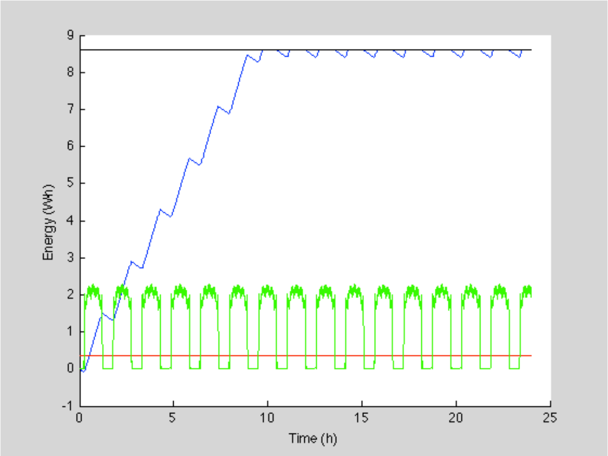


Figure 4.21 Energy plot

The red line shows the profile of the energy required for the subsystems, very small in this case taking into account that only the OBC is on. The black line determines the maximum charge of the batteries, the blue one shows the profile of the satellite's energy and finally, the green profile shows the input power from the sun in W.

The next image shows a realistic case of the satellite. It simulates a full day of the satellites life starting and finishing at 12 am and considering that when starting, the batteries are almost discharged and that we transmit at every opportunity we have, what emulates one worst case.

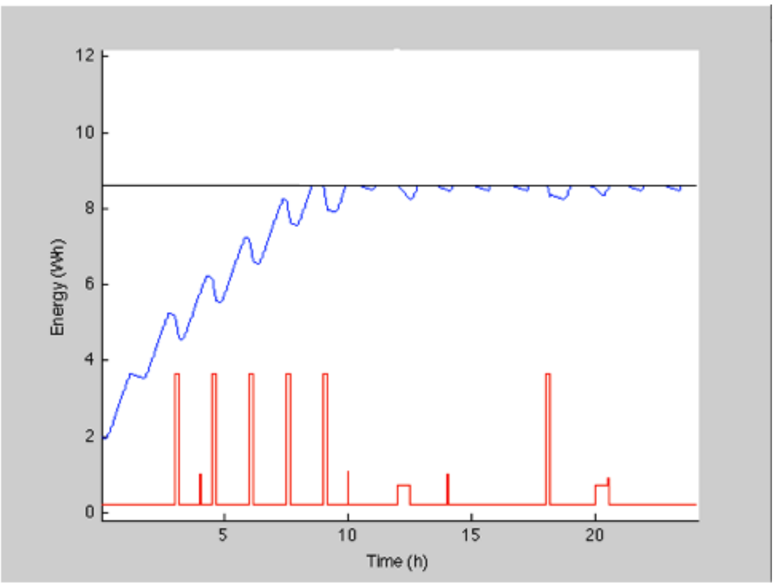


Figure 4.22 Satellite SOC

The simulation shows that considering this case, the satellite will not run out of energy provided none some of the solar panels or batteries get damaged.

Although this simulation can be useful as a first approximation, different scenarios are considered in the next lines also taking into account the amount of data that can be transmitted when facing the ground station. Before these scenarios, a theoretic explanation in presented in order to have a general description, which can be used in other generation of cubesats and not only in this particular one. The results taken from this part will determine the duty cycle of the different payloads considering the amount of data that they generate, the time window to communicate with the ground station and the power consumption, and all these, in the time of an orbit. The different scenarios must give enough information to know if each payload can be executed once every orbit as a worse case, considering the different conditions.

Some of the parameters for the different cases that are invariable are the orbit period and the maximum transmition baud rate. The net transmition rate in the case of the ^{3}CAT is 8000 bps and the orbit period is 90 minutes.

Scenario 1: *Energy worst case*

In this scenario it is considered that the window to communicate with the ground station is the maximum possible, which is about 15 minutes. This scenario will determine if it is possible to execute all the payloads in one orbit and in case it is possible, how much time is still free for redundancy.

First of all, as for the power consumption the simulator is needed, the study about the data is here explained. The amount of data generated is the addition of the data that each device must send to Earth, which amounts up to more than 405 Mb considering the payloads and the monitoring. The first packet sent contains the headers but as it takes less than one second to send it and receive confirmation, it can be despised. As the net baud rate for the transmitter is 8000 bps, it would take about 14 hours to transmit all the information. This huge amount of information comes from considering that the WPT payload uses the maximum hardware capacity, but as it is very configurable, the amount of information can be reduced to 4 Mb, what would need only about 8 minutes to be transmitted.

In terms of power consumption, constraints used are that the batteries are 50% charged after transmitting, which is the worse case as it starts the new orbit when in shadow.

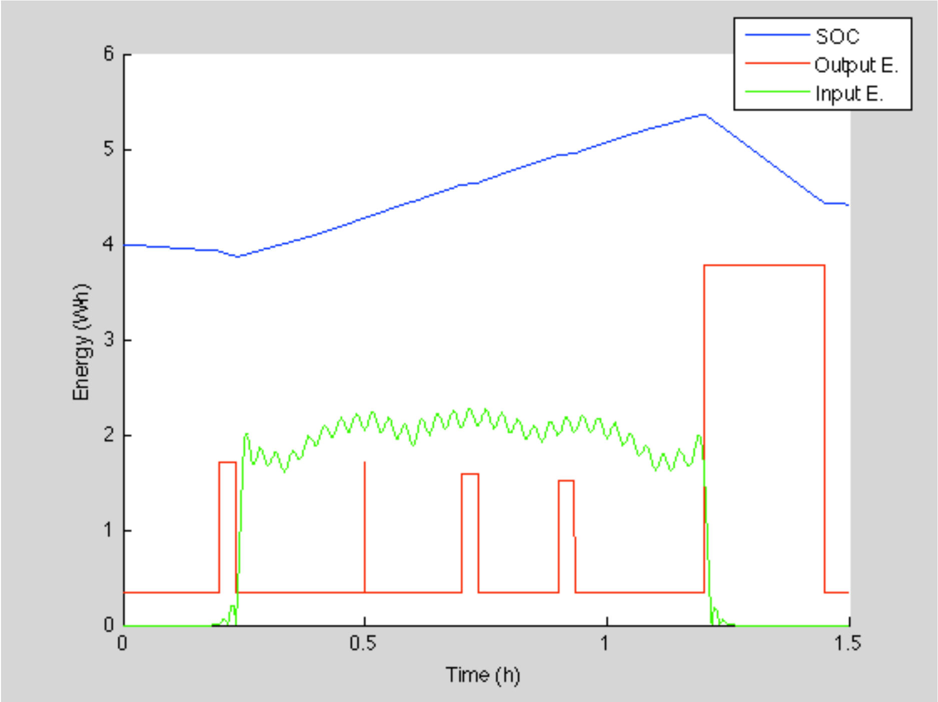


Figure 4.23 Scenario 1

The results from this scenario show that it is possible to execute all the payloads in one orbit, but that it is very risky as the batteries are not able to charge a lot and the simulation only considers the consumption of the main devices.

The conclusion here is that considering the data generated and the time for the communication with the ground station, this option is viable, but when looking at the energy results, they show that it is very risky to make the satellite work this way. Anyway, if the transmittion is done when the satellite faces the sun, it is perfectly possible to operate this way.

Scenario 2: *mean conditions*

This second scenario considers the mean time of communication between the satellite and the ground station, which is about 10 minutes. The objective is the same of the first scenario shown, but with less time of communication.

In terms of time of communication and amount of data to transmit, this scenario has only 2 minutes free to send other information in case there are some errors or to receive information from the Earth to reconfigure some of the parameters or to upload information.

Considering the power demand, the result of the simulation is shown in the next image:

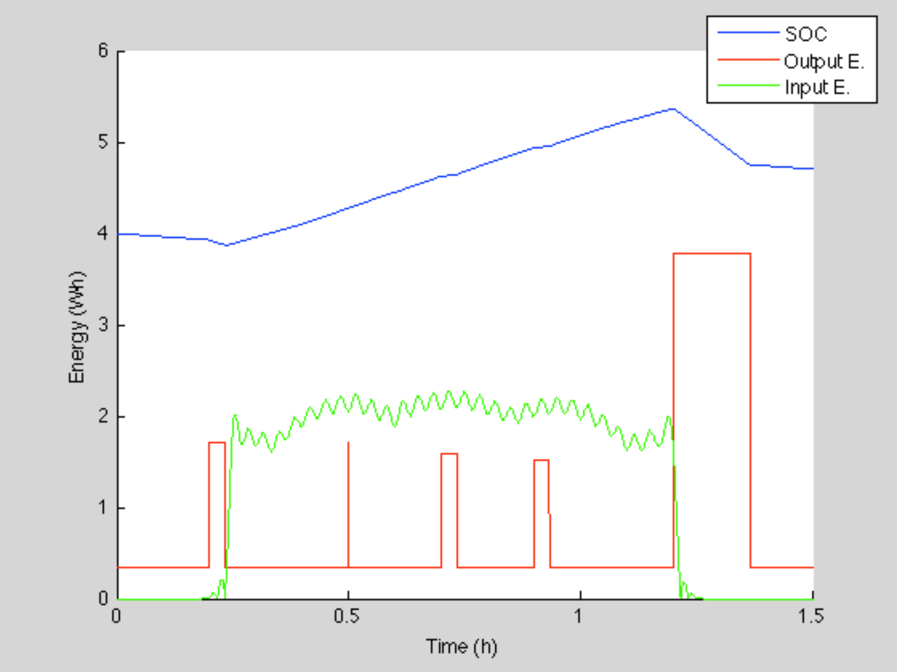


Figure 4.24 Scenario 2

These results show that in terms of power, a communication time of 10 minutes is a lot more suitable than a 15 minutes window.

Scenario 3: *Reduced time access*

This third and last scenario considers the minimum time to communicate with the ground station. This means a window of about 7 minutes.

The objective of this scenario is a bit different from the two above. In the other two cases the objective is to determine if it is possible to run all the devices in one orbit without running out of power and having enough time to transmit all the data generated. This last case wants to define which payloads can run together while satisfying both energy and data limitations only using the 70% of the time windows to communicate with the ground station.

In terms of energy, taking into account the results from the previous scenario, it can be deducted, that no power problems can appear in this scenario.

Remaining that the net baud rate for the communications link is 8000bps and that the maximum time to send information is about 5 minutes, the maximum amount of data that can be transmitted is 2,3 Mb. This means that the information generated by the graphene and WPT payloads is so high that it is impossible to send all of it.

Then, this scenario shows that the OBC needs to take into account the window time to transmit when running some of the payloads and therefore, the data will be separated in different packages (like *.part1, *.part2) and send them when possible using more than one orbit.

4.4.1.1.4 Implications

4.4.1.1.4.1 Portux power consumption

After doing some simulations using the first version of the Energy Manager Simulator, it became to a point where the power consumption of the OBC was too high to respect the power requirements of the system.

At that time, there was the option to move from using PortuxG20 to Stamp9G20 [2], as its consumption was lower than Portux's one. Although it could have worked, the whole team had been working considering the specifications of the Portux. Due to this, the only solution was to reduce the consumption of our OBC. After trying a many different ways to do it, Taskit gave us the solution. It was as easy as disabling the Ethernet microcontroller welding a 10K resistor to the marked place in the image below.

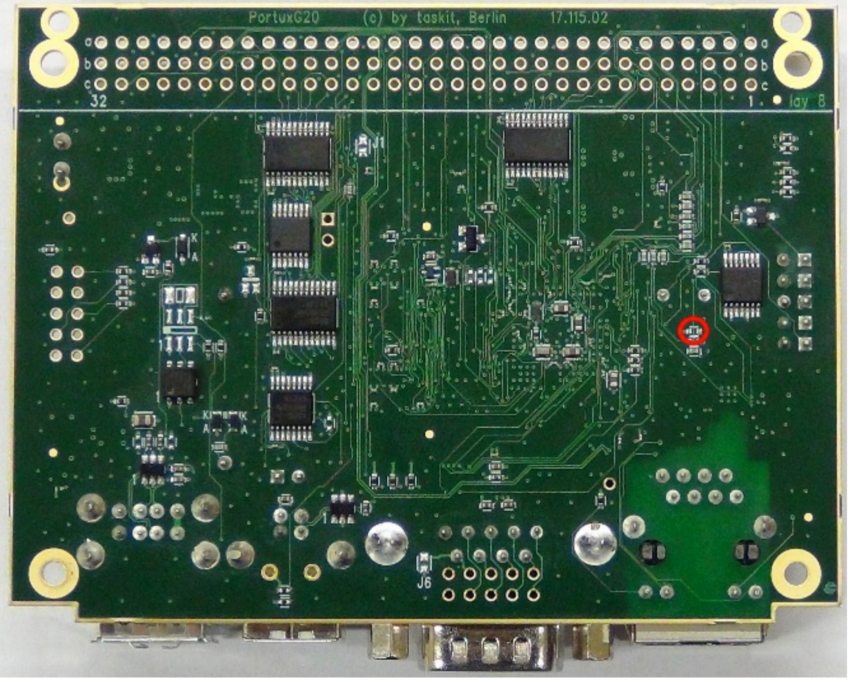


Figure 4.25 Ethernet disabling

After this, the power demand of our OBC was reduced from 400-450mW to 200-250mW.

4.4.1.2 Energy manager

The energy manager is, together with the task scheduler, an entity managed by the On-Board Computer (OBC) responsible for determining the energy available based on the orbital dynamics and current battery state-of-charge. The OBC will then, based on the tasks pending and their priority, decide on whether to perform a task or not. It is based on the simulator but written in C language, as it has to be inside the on-board computer.

4.4.2 Data scheduler simulator

This data scheduler simulator is the tool that should replace the empiric calculations made in the three scenarios above but it is still being developed. The simulator must determine if the amount of data generated in a certain period of time can be sent to Earth at each opportunity of communication or if the scheduler must be very precise due to the large amount of information and the short time of transmission at each orbit.

This simulator has a main function, called dataScheduleStudy(), which is in charge of calling all the functions needed to run the simulation. It has two parameters that can be modified to check different scenarios: the time the simulation starts and the transmission rate of the satellite with the GS (ground station). The first functions called by the program are in charge of getting the information of the desired scenario. This is the block diagram of this first part.

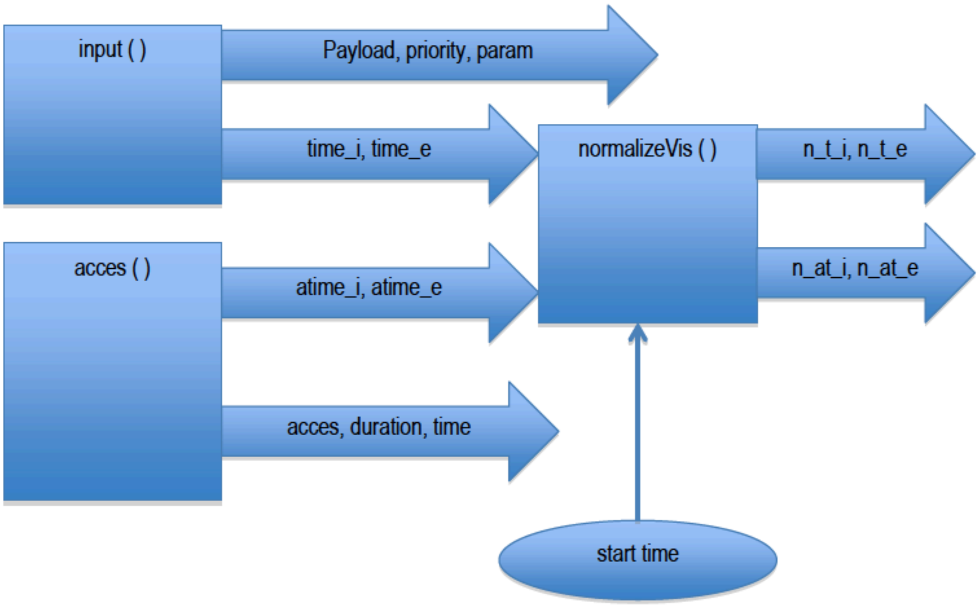


Figure 4.26. Data acquisition block diagram

The function input() is responsible of getting the information about the payloads for that scenario (in which time is each payload turned on and off, and its configuration and priority). The function access() also is in charge of getting information of the scenario, but this one gets the information related to the visibility with the ground station. All the times obtained by these 2 functions have to be converted into numbers to be able to operate with it easily and also to be normalized to a start time. Then, we have a function that computes the amount of data that will be generated by each payload. This function is called dataPayload(). Just after that there is a function that taking into account the chosen transmission rate, the information about all the access times and the information already got from the previous function, which will have computed the amount of data generated by the payloads, will perform the scheduler work. This function, scheduleData(), will be in charge of selecting which payload has to be sent in each access time. The blog diagram of this couple of functions is shown in the next figure.

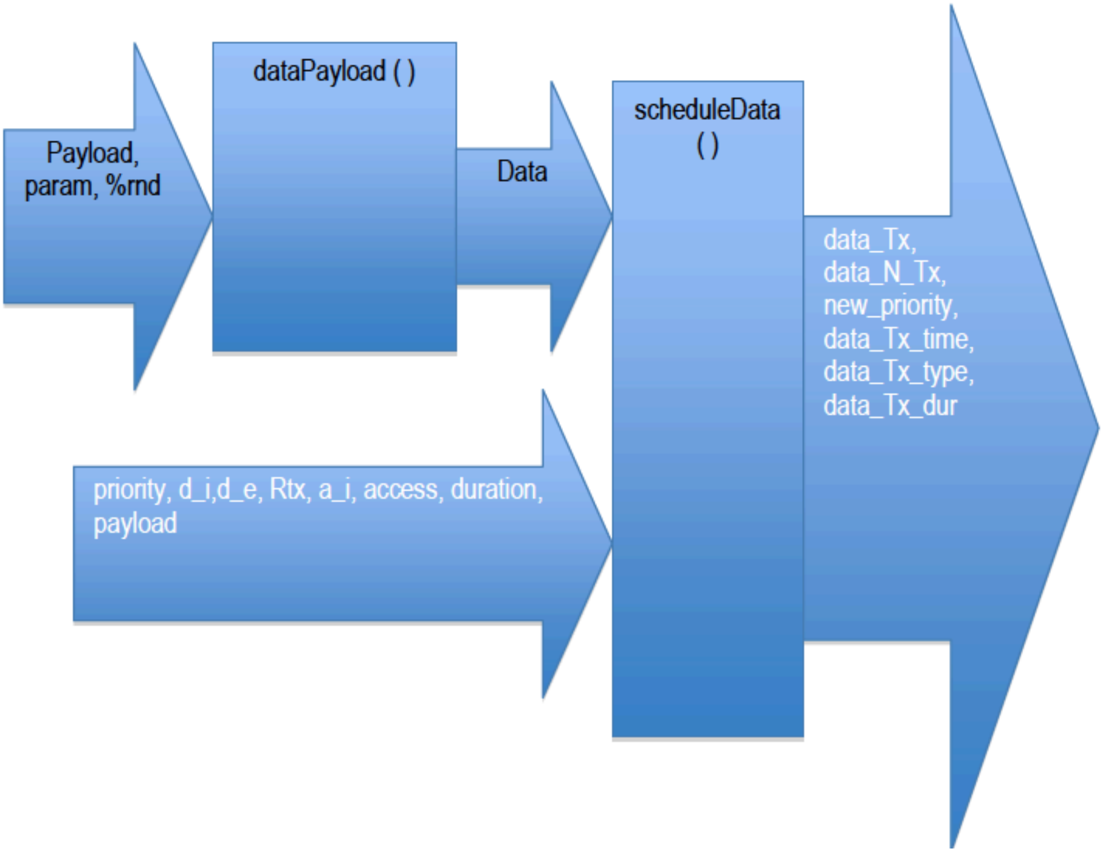


Figure 4.27. Data scheduler bloc diagram

The parameter %rnd used as an input of the function dataPayload(), which has not been mentioned before, is a parameter that gives a random component to the size of the data. This is done in order to simulate 2 things:

- The size of the information generated by certain payloads may vary depending on the information acquired. For example, if the camera takes a photo that is all in the same colour its compression format will make it to be different size from one that contains a wide amount of colours.
- The data transmitted will not always arrive properly so the selective repeat will have to be used. In order to simulate this, it is better to do as if the size of the data is much bigger than normal instead of adding another random component to the access times.

The variables that are returned from the function scheduleData() are only used to represent the results, in order to be easy for somebody who does not know how the program works to do an interpretation of the results. This representation of the results is done using a couple of functions: plotTime() and plotPayload(). The first one is in charge of representing when we have each access time and its duration and the second one will put a mark of a particular colour (depending on the payload). It is useful to represent both informations together in order to see the evolution of the payloads sent taking into account the priority and also to be able to check that there are no problems of a payload that is always occupying the access times.

Bibliography (Chapter 4)

[1]http://www.sandisk.com/products/memory-cards/microsd/extremepro-uhs i/?capacity=8GB

[2] Taskit Stamp9G20 specifications, http://www.taskit.de/en/products/stamp9g20/index.htm

[3] www.cubesatshop.com

[4] A Pedret Otero, UPCSat-1: desarrollo, integración y caracterización de los paneles solares y el sensor óptico **Universitat Politècnica de Barcelona (UPC), Barcelona, 18/05/2012**

[5] http://www.duracelldirect.es/camara-digital/duracell/oem-pno/drf60.html

[6] http://suzymchale.com/ruspace/issorbit.html

[7] *Final Design Review (FDR),* Projecte Avançat d'Enginyeria (PAE) – QP 12/13

[8]Taskit PortuxG20 specifications http://www.taskit.de/en/products/portuxg20/index.htm

[9] **A. Camps,** PAE PICOSATS presentation (Chapter 9, Product Assurance)

[10] **A. Camps,** PAE PICOSATS presentation (Chapter 10, Electromagnetic Compatibility)

[11] Wang Haiying; Hao Zhonghua; Hu Yu; Li Gechen, "Power state prediction of battery based on BP neural network," *Strategic Technology (IFOST), 2012 7th International Forum on* , vol., no., pp.1,4, 18-21 Sept. 2012

[12] Mackowski, M.J.; Martin, D.K., "An energy balance simulation tool for TOMS-EP," *Energy Conversion Engineering Conference, 1996. IECEC 96., Proceedings of the 31st Intersociety* , vol.1, no., pp.641,646 vol.1, 11-16 Aug 1996

[13] Simburger, E.J.; Rumsey, D.; Hinkley, D.; Liu, S.; Carian, P., "Distributed power system for microsatellites," *Photovoltaic Specialists Conference, 2005. Conference Record of the Thirty-first IEEE* , vol., no., pp.822,825, 3-7 Jan. 2005

[14] Sung-Soo Jang, Sung-Hoon Kim, Sang-Ryool Lee, and Jaeho Choi, "Energy Balance and Power Performance Analysis for Satellite in Low Earth Orbit," *J. Astron. Space Sci.* 27(3), 253-262 (2010)

[15] Alberto Sánchez Ávila and Jordi Serra i Font, *"UPCSat-1: Analysis, Design and Breadboarding of a University Picosatellite"*, Master's thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, March 2010.

CHAPTER 5: CONCLUSIONS, LESSONS LEARNED AND FURTHER WORK

5.1. OBJECTIVES AND THEIR ACCOMPLISHMENT

The conclusions that can be taken from this document are those that show that the ^{3}CAT project is perfectly viable. This viability is exposed in the next lines regarding all the parts of this project and the integration of all of them.

From the beginning this project has had a particular objective: make the ^{3}CAT a successful mission. This aim involves a lot of different fields and working channels. On one side there is the kind of work that can be difficultly presented in these lines, as it is to coordinate all the people from the different subsystems, keep in touch with all of them and make periodic meeting to know the state of each device and modify them if necessary. On the other side and being this one the real objective, there is the aim to make this mission succesfull. This target includes two phases, one regarding the results of the other. These two parts are the analysis of the mission and the design to satisfy all the requierements taken from the first part. The analysis of the mission includes learning the Cubesats standard requierements, the study of all the satellite's parts, as it is needed to know how they interfere between them, the analysis of the environmental conditions that the ^{3}CAT will have to survive, and also the functionality of the whole system specially concerning power and transmitions.

The accomplishment of the different objectives presented above are explained in the next lines always facing to the main objective, bring the ^{3}CAT to life. The conclusions regarding the job of working with people from the different subsystems and designing the whole system together to satisfy the needs of all parts is explained in section 5.2, named "Lessons learned".

When writing the conslusions of the second and main objective of this project, I would like to be able to say that the ^{3}CAT is already in orbit and that everything is working fine, but when writing this, the satellite is not finished yet, so the only conclusions about the viability of the mission are taken from the simulations, which assume that each subsystem works properly and that there are no setbacks. Anyway, the results taken from the simulations allow us to be optimistic about the whole system operation. Once all the subsystems and the mission's orbit had been studied and the keypoints had been defined, the main worries fell on the power consumption due to the high energy requiered by some of the subsystems, specially by the communications link and on the lifetime of the different electronic devices in space.

This preoccupations have been dissipated due to the results of this project as they show that it is almost impossible that the satellite runs out of energy (it will never happen as it operates using a safety margin), that the communications with the base station are long enough to transmit all the data generated in each orbit, and taking the results of the study of the cubesat's shielding, that the mission will last long enough to take results from the different payloads.

5.2 LESSONS LEARNED

Here are explained the most important things learned while working on this project. These points can be helpful for further projects since if I had known them before this project, some of the work done would have taken less time and effort.

- **Parallel work.** This is the way to work when having many different subsystems to consider. You need to have all the subsystems being designed at almost the same and taking into account the other subsystems. It is good to have feedbacks from every subsystem to the system integrator as also to the other subsystems, so compatibility issues can be detected before it is to late and a subsystem gets too much closed in specifications due to the others.

- **Coordinating people.** When working in a university research group, it is not easy to put all the team together in a meeting as everyone has different agendas and are working in different things, so it is important to have at least one meeting a week for everyone to explain how his work is developing and to debate about issues that concern everyone.

- **System specifications.** Before starting an integration project, the most important thing is to have all the specifications of the whole system and of every other part very clear. Otherwise, it is very difficult to put all the work together without making mistakes due to the lack of knowledge. When working on the project, one always learns new things, but they must be things to improve the system, not to change it because you didn't know that before.

- **Taking decisions.** In some of the points a project, one has to take decisions and this is not something easy as you can go wrong or you don't have many information about that. However, when a decision is postponed, it only delays the development of the system, so the decisions must be taken although you may be wrong.

- **Timeline and deadlines.** As happens with many things in life, people usually work better under pressure, so at the beginning of a project it is advisable to define a timeline and deadlines as people will work harder to achieve their objective in that period of time, otherwise people relax and take a lot of time to develop their systems.

5.3 FURTHER WORK

From now on, the work to be done in the ^{3}CAT is to start the real integration, until now what has been done is design integration, but now it is time to test the subsystems and assemble all the satellite.

This sounds not so much work, but when things are manufactured and mounted are not always as perfect as expected in the simulations or in the models, so the final integration of the whole system regarding physical assembly and operation can take longer than expected and some parts may need to be redesigned.

When finishing this text, some more work to do was found. A very important point is to perform an electromagnetic compatibility analysis in order to avoid any interference between the different devices. Also, the way to hold the OBC inside the satellite must be re-designed as it was driell in order to math the columns of the structure, but after doing it, the board didn’t work anymore. Another field where more work should be done is the power and data simulators, which should be mixed in orther to have only one application to study the viability of the project.

Appendix A. Energy simulator code

```
clc
clear all

% LOAD .CSV FILE FROM STK

FILENAME = uigetfile('*.csv');

%fid = fopen( fullfile('SolarPanelPowerResults',FILENAME) );

fid = fopen( FILENAME );

DATA = textscan(fid, '%s %f %d', 'Delimiter',',','HeaderLines',1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CELL MANIPULATION

TIME = DATA{1};
POWER = DATA{2};
INTENSITY = DATA{3};

numSolarPanelGroups = 6;
timeStep = 60; % 60 sec
simStartTime = char(TIME(1,:)); % simulation start time
simStopTime = char(TIME(end,:)); % simulation stop time

tf = ~cellfun( 'isempty',strfind(TIME,simStartTime) );
cellIdx = find(tf);

samples = cellIdx(2);

% FIND SOLARPANELGROUPS
tf = ~cellfun( 'isempty',strfind(TIME,simStartTime) );
cellIdx = find(tf);

% [   lateral   ]
% [   deploy    ]
% [    baix     ]
% [   cellsat   ]
% [  lateral2   ]
% [    dalt     ]

SPANEL_POWER = zeros(numSolarPanelGroups, samples);

for i=1:numSolarPanelGroups

    SPANEL_POWER(i,:) = POWER(cellIdx(i):(cellIdx(i+1)))';

end
% DETREND

totalAvPower = sum(SPANEL_POWER);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Psolin=abs(interp(totalAvPower,60));

%%%%%%%%%%%%%%% CONSUMPTION COMMS %%%%%%%%%%%%%%%%%%

%BEACON
consum_beacon = 165;
consum_lo = 120;
consum_mixer_beacon = 340;


%TRANSMISSION
consum_board_tx = 99;
consum_ampli_tx = 3000;
consum_mixer_tx = 340;
temps_tx = 600;

%RECEPTION

consum_board_rx = 55.5;
```

```
consum_lna = 21;
temps_rx = 15;

%%%%%%%%%%%%%% CONSUMPTION PAYLOADS %%%%%%%%%%%%%%%%%

consum_geiger = 150;

consum_camera = 15;
temps_camera = 5;
fotos = 1;

consum_mems = 100;
temps_mems = 120;

consum_coils = 300;
temps_coils = 0.01;

consum_grafe = 175;
temps_grafe= 30;

consum_attitude = 500;
temps_attitude = 1800;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 potencia_beacon = consum_beacon + consum_lo + consum_mixer_beacon;

 potencia_tx = consum_board_tx + consum_ampli_tx + consum_mixer_tx;

 potencia_rx = consum_board_rx + consum_lna;

%%%%%%%% BATERIES MAX CHARGE AND DEGRADATION %%%%%%%%

cmax_wat=8.6;
cmax_degradacio=8.6;
rendiment_conversors=0.9; %EFFICIENCY

T=3600*input ('Total time in hours: ');
tini=60*input('Position: ');
e_inici=input('Initial energy: ');
cicles = input ('Batteries cycle: ');

tt=tini:1:T-1;
pin=1000*rendiment_conversors*Psolin(tini+1:T);
Pout=zeros(1,T-tini);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

degradacio= cicles; %sera una funcio depenent de la caracteritzacio

if degradacio ~= 0
    cmax_degradacio = cmax_wat*(1-degradacio/100);
end

%%%%%%%%%%%%%% PORTUX %%%%%%%%%%%%%%%%%

estat_portux = 'standby';%input ('\nPortux state (standby, power down, on, full load):
','s');

if (strcmp('standby',estat_portux))
       consum_portux = 224;
    else if (strcmp('power down',estat_portux))
       consum_portux = 141;
    else if (strcmp('on',estat_portux))
      consum_portux = 450 + consum_geiger;
    else if (strcmp('full load',estat_portux))
     consum_portux = 630;
        end
        end
        end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
accio = 'inici';
Pout(1:T-tini) = consum_portux;

while true

    tinici=0;
```

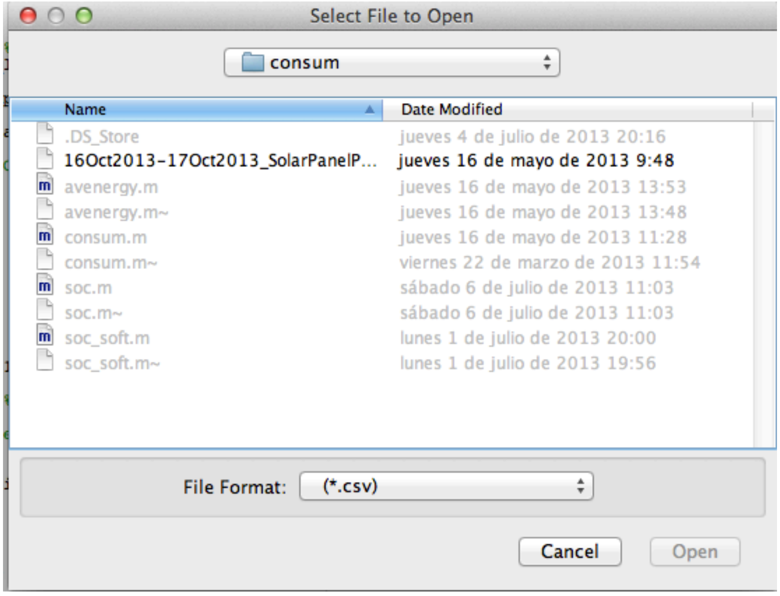
```
    accio = input ('\nSubsystem to actiavate (tx, rx, foto, coils, grafe, mems, end):
','s');

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Pouttx=0;
    Pout1tx=0;
    Pout2tx=0;
    Pout3tx=0;

    Poutrx=0;
    Pout1rx=0;
    Pout2rx=0;
    Pout3rx=0;

    Poutf=0;
    Pout1f=0;
    Pout2f=0;
    Pout3f=0;
    Pout4f=0;

    Poutc=0;
    Pout1c=0;
    Pout2c=0;
    Pout3c=0;

    Poutg=0;
    Pout1g=0;
    Pout2g=0;
    Pout3g=0;

    Poutm=0;
    Pout1m=0;
    Pout2m=0;
    Pout3m=0;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      fi=strcmp(accio,'end');
    if (fi==1)
        break;
    end

    tinici = tini + 3600*input ('\nSubsystem start time in hours from simulation start
time: ');


    switch accio

        case 'tx'
            ttx=tinici;
            P = potencia_tx;

            t1=tini:1:ttx-1;
            t2=ttx:1:(ttx+temps_tx-1);
            t3=(ttx+temps_tx):1:T-1;
            t=[t1,t2,t3];

            Pout1tx(1:ttx-tini)=0;
            Pout2tx(1:temps_tx) = P;
            Pout3tx(1:(T-(ttx+temps_tx)))=0;

           Pouttx = [Pout1tx,Pout2tx,Pout3tx];


        case 'rx'
            trx=tinici;
            P = potencia_rx + potencia_beacon;

            t1=0:1:trx-1;
            t2=trx:1:(trx+temps_rx-1);
            t3=(trx+temps_rx):1:T-1;
            t=[t1,t2,t3];

            Pout1rx(1:trx)=0;
            Pout2rx(1:temps_rx) = P;
            Pout3rx(1:(T-(trx+temps_rx)))=0;

            Poutrx = [Pout1rx,Pout2rx,Pout3rx];
```

```
case 'attitude'
     tat=tinici;
     P = consum_attitude;

     t1=0:1:tat-1;
     t2=tat:1:(tat+temps_attitude-1);
     t3=(tat+temps_attitude):1:T-1;
     t=[t1,t2,t3];

     Pout1at(1:tat)=0;
     Pout2at(1:temps_attitude) = P;
     Pout3at(1:(T-(tat+temps_attitude)))=0;

     Poutrx = [Pout1at,Pout2at,Pout3at];

 case 'foto'
     tf=tinici;
     P = potencia_rx - consum_lna + potencia_beacon + consum_camera;

     t1=0:1:tf-1;
     t2=tf:1:(tf+temps_attitude-1);
     t3=(tf+temps_attitude):1:(tf+temps_attitude+(fotos*temps_camera)-1);
     t4=(tf+temps_attitude+(fotos*temps_camera)):1:T-1;
     t=[t1,t2,t3,t4];

   Pout1f(1:tinici)=0;
   Pout2f(1:temps_attitude) = consum_attitude;
   Pout3f(1:(fotos*temps_camera)) = P;
   Pout4f(1:(T-(tinici+temps_attitude+(fotos*temps_camera))))=0;

   Poutf = [Pout1f,Pout2f,Pout3f,Pout4f];


 case 'coils'
     tc=tinici;
     P = potencia_rx - consum_lna + potencia_beacon + consum_coils;

     t1=0:1:tc-1;
     t2=tc:1:tc+temps_coils-1;
     t3=(tc+temps_coils):1:T-1;
     t=[t1,t2,t3];

     Pout1c(1:tc)=0;
     Pout2c(1:temps_coils) = P;
     Pout3c(1:(T-(tc+temps_coils)))=0;

     Poutc = [Pout1c,Pout2c,Pout3c];

 case 'grafe'
     tg=tinici;
     P = potencia_rx - consum_lna + potencia_beacon + consum_grafe;

     t1=0:1:tg-1;
     t2=tg:1:(tg+temps_grafe-1);
     t3=(tg+temps_grafe):1:T-1;
     t=[t1,t2,t3];

     Pout1g(1:tg)=0;
     Pout2g(1:temps_grafe) = P;
     Pout3g(1:(T-(tg+temps_grafe)))=0;

     Poutg = [Pout1g,Pout2g,Pout3g];


 case 'mems'
     tm=tinici;
     P = potencia_rx - consum_lna + potencia_beacon + consum_mems;

     t1=0:1:tm-1;
     t2=tm:1:(tinici+temps_mems-1);
     t3=(tm+temps_mems):1:T-1;
     t=[t1,t2,t3];
```

```
            Pout1m(1:tm)=0;
            Pout2m(1:temps_mems) = P;
            Pout3m(1:(T-(tm+temps_mems)))=0;

            Poutm = [Pout1m,Pout2m,Pout3m];



    end
    Pout=Pout + Pouttx + Poutrx + Poutf + Poutc + Poutg + Poutm;

end

Pout_wat=Pout./1000;
%%State of charge
SOC=zeros(1,T-tini);


pin_wat=pin./1000;
%
 for i=1:T-tini
     if i==1
         SOC(i)=e_inici+(pin_wat(i)-Pout_wat(i))./3600;
     else
         SOC(i)=SOC(i-1)+(pin_wat(i)-Pout_wat(i))./3600;
     end

     if SOC(i)>cmax_degradacio,

          if Pout_wat(i)~=consum_portux/1000,
             %%%instant power%%%
             SOC(i)=SOC(i-1)+pin_wat(i)-Pout_wat(i);
             if SOC(i)<cmax_degradacio,
                 SOC(i)=cmax_degradacio-(pin_wat(i)-Pout_wat(i))./3600;
             end
             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          else
            SOC(i)=cmax_degradacio+pin_wat(i);
          end
     end
 end

hold
plot(tt/3600,SOC);
plot(tt/3600,cmax_wat,'g')
plot(tt/3600,Pout_wat,'r');
plot(tt/3600,cmax_degradacio,'k')
```

Appendix B. Energy simulator instructions

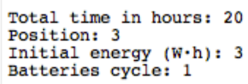
When running the simulator, the first window that will open asks for a file. This file must be a csv file containing the values taken from STK relating to the input energy from the Sun.



Name	Date Modified
.DS_Store	jueves 4 de julio de 2013 20:16
16Oct2013-17Oct2013_SolarPanelP...	jueves 16 de mayo de 2013 9:48
avenergy.m	jueves 16 de mayo de 2013 13:53
avenergy.m~	jueves 16 de mayo de 2013 13:48
consum.m	jueves 16 de mayo de 2013 11:28
consum.m~	viernes 22 de marzo de 2013 11:54
soc.m	sábado 6 de julio de 2013 11:03
soc.m~	sábado 6 de julio de 2013 11:03
soc_soft.m	lunes 1 de julio de 2013 20:00
soc_soft.m~	lunes 1 de julio de 2013 19:56

File Format: (*.csv)

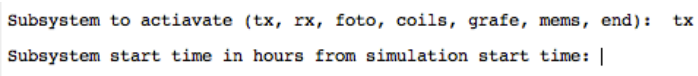
After opening the file, the Matlab command window opens and asks for different parameters to introduce.
The first part is to define how long is going to be the simulation and characteristics of the general state and degradation of the satellite.



```
Total time in hours: 20
Position: 3
Initial energy (W·h): 3
Batteries cycle: 1
```

The first input is the duration of the simulation, the second one the actual position of the satellite in its orbit, the next one is the current energy in the batteries and finally, the number of complete cycles of the battery to calculate its degradation.

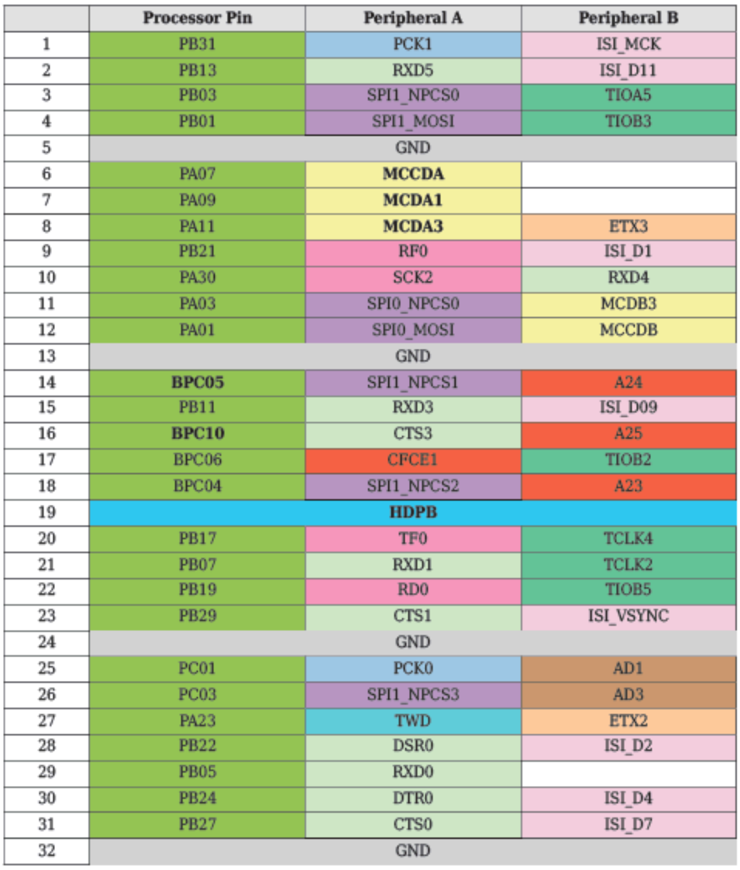
The next questions that appear are now related to the subsystems that the OBC is going to activate. It asks for the payload to activate and when to do it.



```
Subsystem to actiavate (tx, rx, foto, coils, grafe, mems, end):  tx

Subsystem start time in hours from simulation start time: |
```

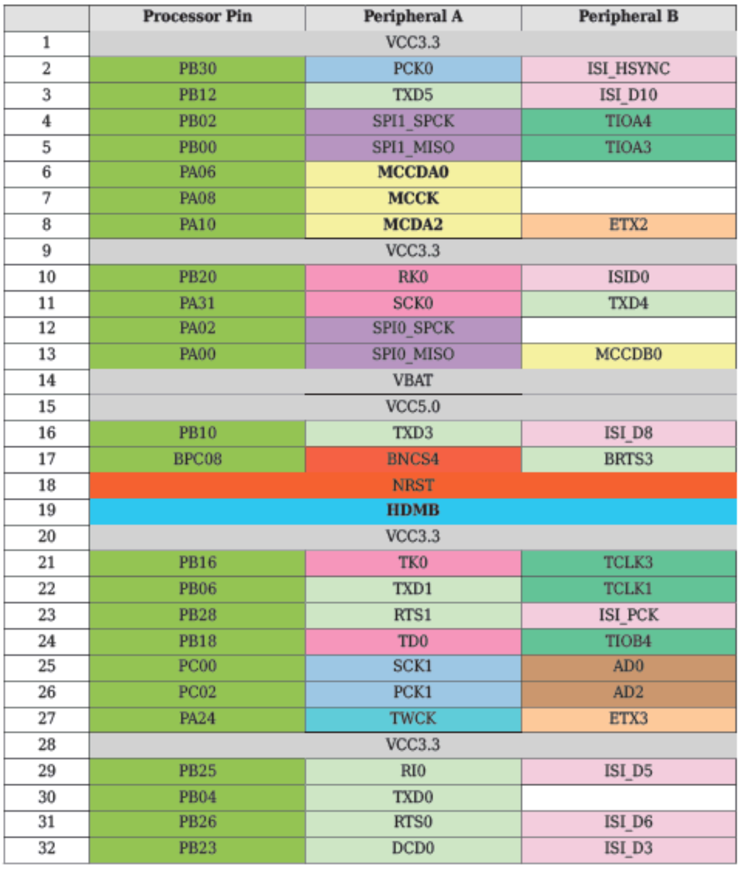
You can enter as many payloads as you want and when finished, write 'end' and the simulation will start.

Appendix C. Portux pins



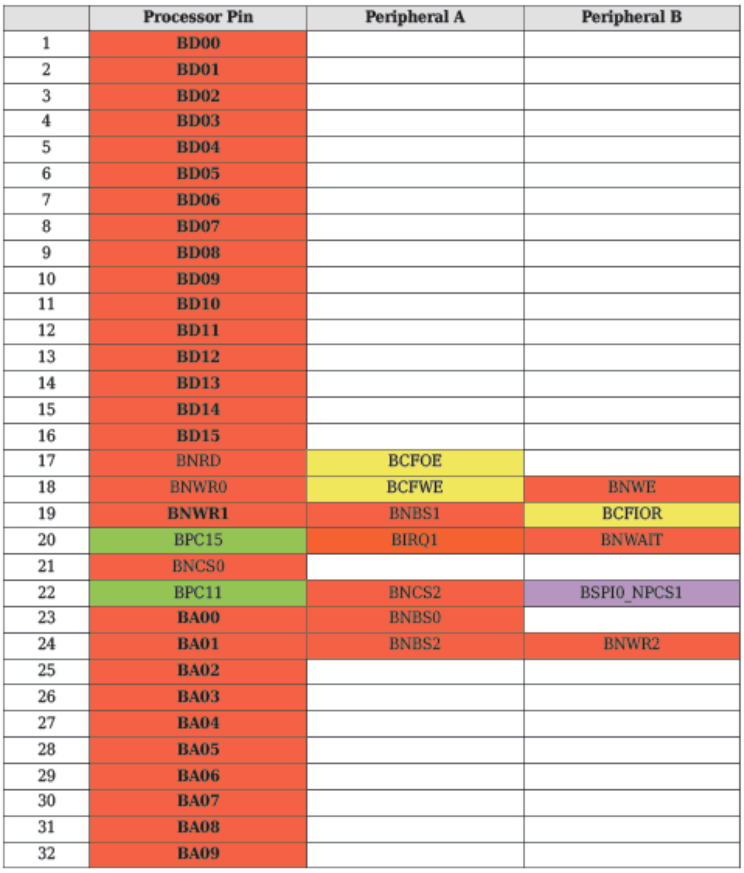
	Processor Pin	Peripheral A	Peripheral B
1	PB31	PCK1	ISI_MCK
2	PB13	RXD5	ISI_D11
3	PB03	SPI1_NPCS0	TIOA5
4	PB01	SPI1_MOSI	TIOB3
5	GND		
6	PA07	**MCCDA**	
7	PA09	**MCDA1**	
8	PA11	**MCDA3**	ETX3
9	PB21	RF0	ISI_D1
10	PA30	SCK2	RXD4
11	PA03	SPI0_NPCS0	MCDB3
12	PA01	SPI0_MOSI	MCCDB
13	GND		
14	**BPC05**	SPI1_NPCS1	A24
15	PB11	RXD3	ISI_D09
16	**BPC10**	CTS3	A25
17	BPC06	CFCE1	TIOB2
18	BPC04	SPI1_NPCS2	A23
19	**HDPB**		
20	PB17	TF0	TCLK4
21	PB07	RXD1	TCLK2
22	PB19	RD0	TIOB5
23	PB29	CTS1	ISI_VSYNC
24	GND		
25	PC01	PCK0	AD1
26	PC03	SPI1_NPCS3	AD3
27	PA23	TWD	ETX2
28	PB22	DSR0	ISI_D2
29	PB05	RXD0	
30	PB24	DTR0	ISI_D4
31	PB27	CTS0	ISI_D7
32	GND		

Table D.1. Pin Assignment PXB Row A



	Processor Pin	Peripheral A	Peripheral B
1	VCC3.3		
2	PB30	PCK0	ISI_HSYNC
3	PB12	TXD5	ISI_D10
4	PB02	SPI1_SPCK	TIOA4
5	PB00	SPI1_MISO	TIOA3
6	PA06	**MCCDA0**	
7	PA08	**MCCK**	
8	PA10	**MCDA2**	ETX2
9	VCC3.3		
10	PB20	RK0	ISID0
11	PA31	SCK0	TXD4
12	PA02	SPI0_SPCK	
13	PA00	SPI0_MISO	MCCDB0
14	VBAT		
15	VCC5.0		
16	PB10	TXD3	ISI_D8
17	BPC08	BNCS4	BRTS3
18	NRST		
19	**HDMB**		
20	VCC3.3		
21	PB16	TK0	TCLK3
22	PB06	TXD1	TCLK1
23	PB28	RTS1	ISI_PCK
24	PB18	TD0	TIOB4
25	PC00	SCK1	AD0
26	PC02	PCK1	AD2
27	PA24	TWCK	ETX3
28	VCC3.3		
29	PB25	RI0	ISI_D5
30	PB04	TXD0	
31	PB26	RTS0	ISI_D6
32	PB23	DCD0	ISI_D3

Table D.2. Pin Assignment PXB Row B



	Processor Pin	Peripheral A	Peripheral B
1	BD00		
2	BD01		
3	BD02		
4	BD03		
5	BD04		
6	BD05		
7	BD06		
8	BD07		
9	BD08		
10	BD09		
11	BD10		
12	BD11		
13	BD12		
14	BD13		
15	BD14		
16	BD15		
17	BNRD	BCFOE	
18	BNWR0	BCFWE	BNWE
19	BNWR1	BNBS1	BCFIOR
20	BPC15	BIRQ1	BNWAIT
21	BNCS0		
22	BPC11	BNCS2	BSPI0_NPCS1
23	BA00	BNBS0	
24	BA01	BNBS2	BNWR2
25	BA02		
26	BA03		
27	BA04		
28	BA05		
29	BA06		
30	BA07		
31	BA08		
32	BA09		

Table D.3. Pin Assignment PXB Row C

Appendix D. Pin assignment

EPS

1 ADC	A25
POL	B2,B6,B13,B12,B17,B26,B29,B23,B27
RX,TX	A10,B11
EPS-Comms	A6
Voltage	B8, C8

COMMS(5V,3.3)

2 enable	B2,B6
1 SPI	A3,A4,B4,B5
Beacon inputs EPS	A6
COMMS-EPS	B8
Data-peltier	A7
Gnd-peltier	A8
Vcc peltier	B7
Antenna deployment	A1

WPT+Graphene(3.3,5)

1 SPI	A4,A14,B4,B5
1 enable	B12
Digital	A12

ATTITUDE(3.3,5)

Tx, Rx	A2,B3
2 Enable	B26,B27

CAMERA (5V)

Tx,Rx	A15,B6
1 Enable	B17

MEMS(5,3.3,1.8)

Tx,Rx	A29,B30
1 enable	B29

GEIGER(5V)

1 Counter	B24
1 enable	B23