

# UnB On-Board Computer Prototype for CubeSats

G. Silva Lionço(1), G. Santilli (2), L. Aguayo (3).

(1) Universidade de Brasília  
Faculdade Gama Brasília, Brazil  
Phone: +55 61 9 9227-8739, Mail: [guilherme.lionzo@gmail.com](mailto:guilherme.lionzo@gmail.com)  
(2) Universidade de Brasília  
Faculdade Gama Brasília, Brazil  
Phone: +55 61 9 8355-0174, Mail: [santilli@aerospace.unb.br](mailto:santilli@aerospace.unb.br)  
(3) Universidade de Brasília  
Faculdade Gama Brasília, Brazil  
Phone: +55 61 31078901, Mail: [aguayo@unb.br](mailto:aguayo@unb.br)

**Abstract:** Researchers of the University of Brasília (UnB) are studying the feasibility of the deployment of 3U CubeSat missions as a technology demonstrator. Some studies are already being carried out, in order to offer solutions to this future mission. The present research is aimed at the construction of an Onboard Computer (OBC) for this future mission. During the development and prototyping of the OBC, it was used the co-design methodology, which allowed the hardware and software development at the same time. For the theoretical project, it was chosen the microcontroller and other devices to compose the OBC's hardware. For the embedded software, the FreeRTOS was defined as the Real-Time Operating System. During the protoboard test, it was possible to verify: the consumption of the microcontroller; modes of operation of the embedded software; the acquisition and data storage; etc. It was concluded that the use of the TI MSP432 is a great choice for low-power and intermediate performance scenarios. The use of FreeRTOS as a real-time operating system for low memory systems, as well as the use of watchdog utilization at software level has been ratified. The OBC is still in the prototyping process and possibly will be completed in the coming months.

## 1. INTRODUCTION

The feasibility of the deployment of 3U CubeSat missions is under study at the University of Brasília (UnB). These missions are aimed to validate concepts and perform system/subsystem/component testing for future proof-of-concepts prototypes. Initial subsystem requirements for these missions are (a) use of optical cameras for polar shields monitoring, (b) use of a Pulsed Plasma Thruster (PPT) for orbital control studies and (c) the use of an accelerometer to provide a mapping of the Earth gravitational field.

Some studies are already ongoing and all missions will require an Onboard Computer (OBC), a mandatory subsystem responsible for the control, management, and monitoring of all relevant signals needed for the correct operation of the CubeSat. The following text comprises the details of the development of UNB OBC, from its requirements to the hardware and software specifications, as well as some results of the first prototype.

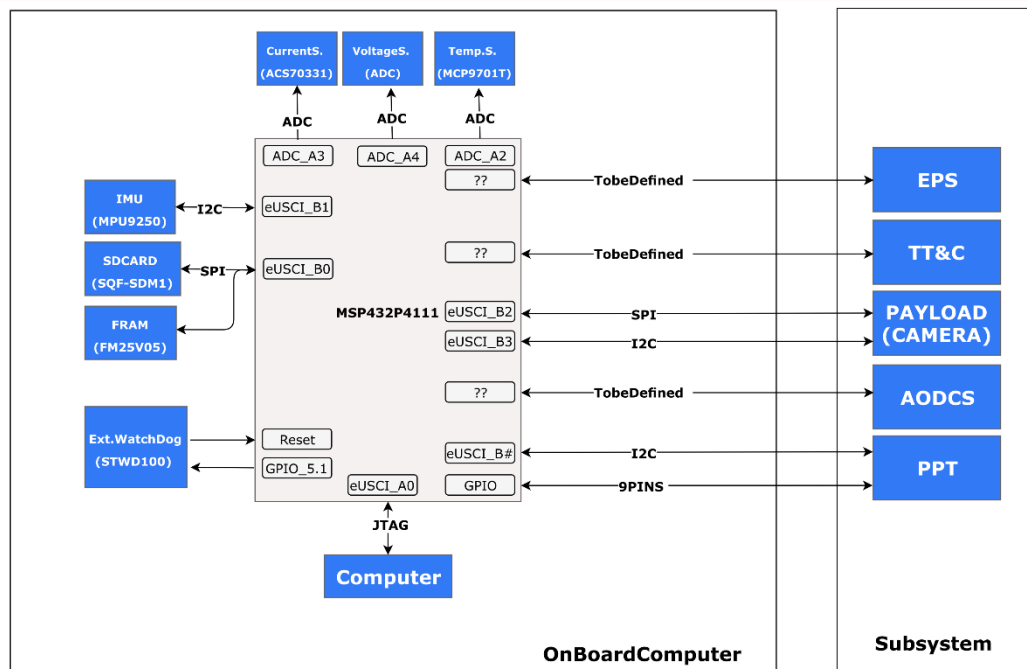
## 2. OBC REQUIREMENTS FOR 3U CUBESAT MISSION

Among the primary requirements for the initial phase of the OBC subsystem project, some are depicted as follows. The list is not exhaustive but contain the major functions to be performed at this stage of prototyping.

- To control an optical camera based on CMOS (*Complementary Metal-Oxide-Semiconductor*) technology;
- To provide control for a Pulsed Plasma Thruster;
- To control and process signals from an inertial sensor;
- To guarantee a high degree of reliability, even using commercial off-the-shelf (COTS) components;
- To store useful data in a non-volatile memory, for adequate transmission to a ground station;
- To have an anti-locking system;
- Change operation modes according to the energy availability.

### 3. OBC HARDWARE AND SOFTWARE ARCHITECTURE

A *co-design* methodology was used for the development of OBC, i.e., there was a simultaneous and interactive development of hardware and software. Figure 1 illustrates the general architecture, as well the interfaces with other subsystems. The next subsections provide details on the hardware subsystems and software functionalities.



**Figure 1** – Block diagram of the OBC, with interfaces.

#### 3.1. Hardware Architecture with COTS Components

The first step was the assessment of different options for the microcontroller, as it is the component that imposes the main restrictions on the OBC performance. Selection criteria has considered: low power consumption, presence of an analog-to-digital converter (ADC), standard communication interfaces such as General-Purpose Input/output (GPIO), serial peripheral interface (SPI) and processing speed (clock rate). Usual COTS components were considered, and to reduce the time for the prototyping, we also required that a development kit was ready to be acquired from the corresponding manufacturer.

Among a list of candidates, the final selection was Texas Instruments MSP432P4111, due to its low power consumption and satisfactory speed performance. Its power consumption

is about 520  $\mu\text{W}/\text{MHz}$  and has memory size compatible with the mission requirements. Furthermore, it has 18 different operation modes, timers, ADCs, several I/O pins and a block of real-time clock [1].

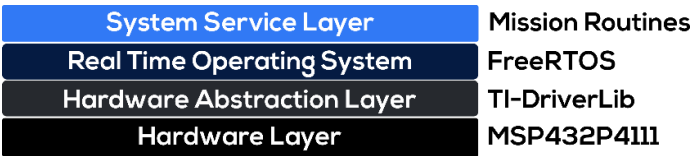
The second step was the dimensioning of the memory subsystem, responsible to store data from OBC processing, and composed by two subunits. The first stores telemetry and payload data, and the second stores a backup of the embedded software. For the estimative of data requirements, the team has used the parameters of *SWISSCube* mission [2]: an amount of 176 MB of data per day, 10 MB from telemetry and 166 MB from the payload (optical camera). The number of ground stations has not been defined yet, so it was chosen the maximum of data addressing that the microcontroller supports, which is 4 GB. For the backup of embedded software, a Ferroelectric RAM (FRAM) memory was chosen due to its resistance to TID and SEU effects [3]. The set size was twice the microcontroller SRAM memory, resulting in 512KB.

Finally, it was chosen COTS components to be used as peripherals that, like the storage unit, are intended to assist the microprocessor in meeting OBC requirements. This unit is basically composed of four components: ACS70331 current sensor [4]; MPU9250 inertial sensor [5], MCP9701T temperature sensor [6], STWD100 external watchdog [7].

In addition to the microprocessor, memory and peripheral subsystems, the OBC unit has two main interfaces, one for debugging and other for communicating with other satellite subsystems. The debug interface uses the Joint Test Action Group (JTAG) and Serial Wire Debug (SWD) pins [8]. The second interface is reserved for communication with satellite subsystems, and is made through the 16bit ISA bus, used in PC104 standard boards [9].

### 3.2. Software Architecture

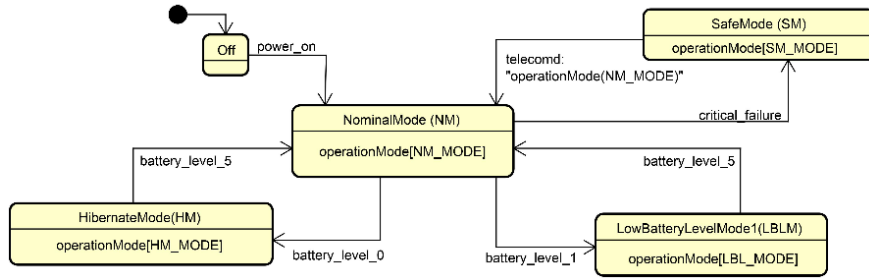
As depicted in Figure 2, a layered approach architecture was defined, in order to provide better software usability and maintenance. Mission parameters may be set at the top layer, System Service Layer (SSL). The Hardware Abstraction Layer (HAL) is performed by the TI-DriverLib [10] and the FreeRTOS was used to meet the requirements of Hard RTOS [11].



**Figure 2** – Layered software architecture.

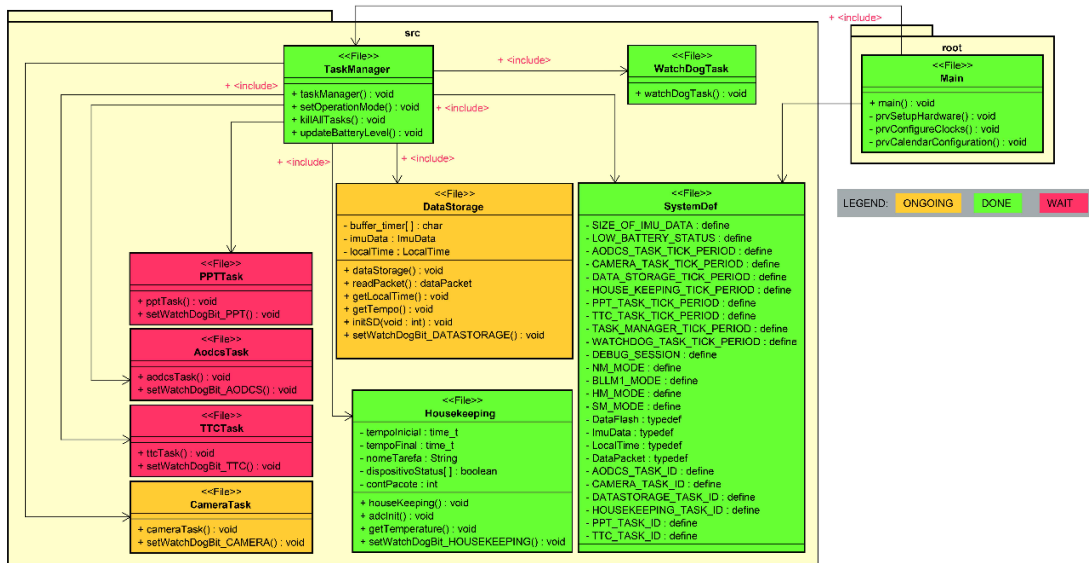
Here we focus only on the SSL, the layer with the main development effort. An UML extension, FuncionalC [12], allows the modeling of systems using C language, the one chosen for the code writing and testing. It is worth mentioning that, at this stage of prototyping, the pre-launching and deployment stages were not considered.

From the software documentation perspective, two major representations were produced: a State Machine Diagram and a File Diagram. The first describes all four modes of operation of the OBC, as depicted at Figure 3: **Nominal Mode**, **Safe Mode**, **Battery Low Level Mode** and **Hibernate Mode**.



**Figure 3** – System Service Layer State Machine Diagram.

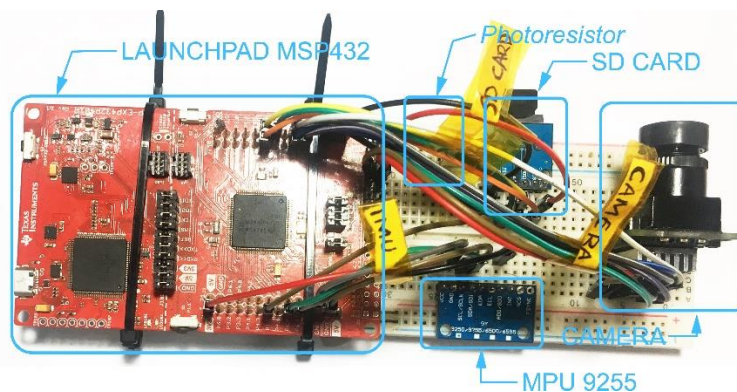
The File Diagram has 9 routines, presented in Figure 5: one for control (**TaskManager**), one for data collection (**HouseKeeping**), one for data storage (**DataStorage**), one for software locking control (**WatchDogTask**), and 5 to handle CubeSat subsystems functionalities, such as telemetry, tracking and command (**TT&C**), among others.



**Figure 4** – System File Diagram. Green modules are completed and tested.

## 4. RESULTS

While the OBC hardware was under construction, the MSP432 LaunchPad kit was used to develop all the main software features to be embedded. A photoresistor was used to simulate the battery level of the CubeSat EPS, as the incidence of light interferes with the amount of energy stored in the EPS. Figure 5 shows the prototype for software development with COTS modules (SD Card, MPU9255, photoresist) and the camera.



**Figure 5** – Protoboard with COTS components and LaunchPad.

### 4.1. Data Acquisition and Storage

The data acquisition and storage service are performed by the HouseKeeping and DataStorage tasks, respectively. During the tests performed, the data were saved in ASCII format to facilitate debugging. Figure 6 shows the telemetry file saved on the SD Card.

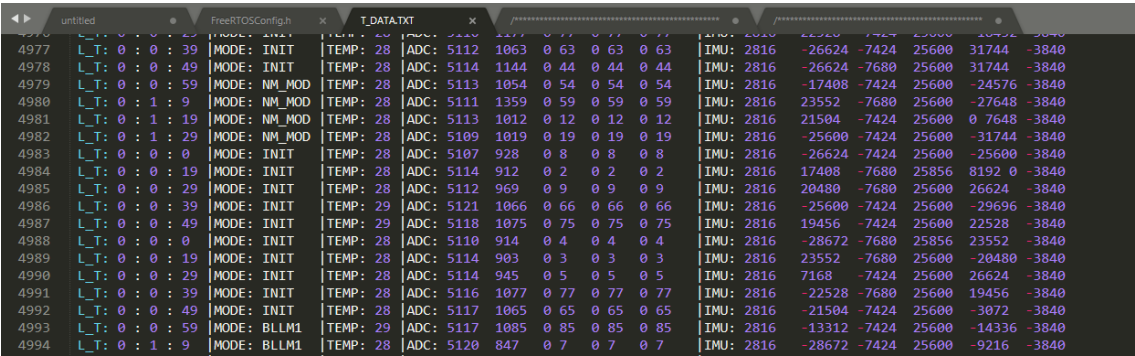


Figure 6 – Telemetry data stored on memory card.

### 4.2. Consumption

Consumption tests were performed using Code Composer Studio's EnergyTracer tool. This tool measures the current being consumed on the JTAG / SW bus, so it allows you to calculate the current consumed by the modules / sensors powered by LaunchPad. There were three five-minute test batteries, one for each state.

Comparing the nominal state with the hibernate state, it is observed that there was a savings of over 40%, increasing the battery life by two days. Figure 7 shows the comparison in graphical form, hibernation in blue and nominal in yellow.

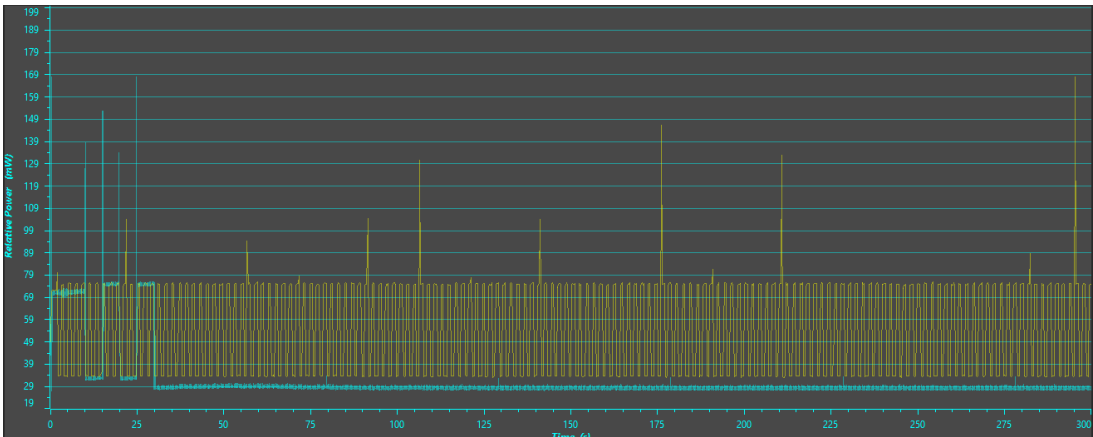
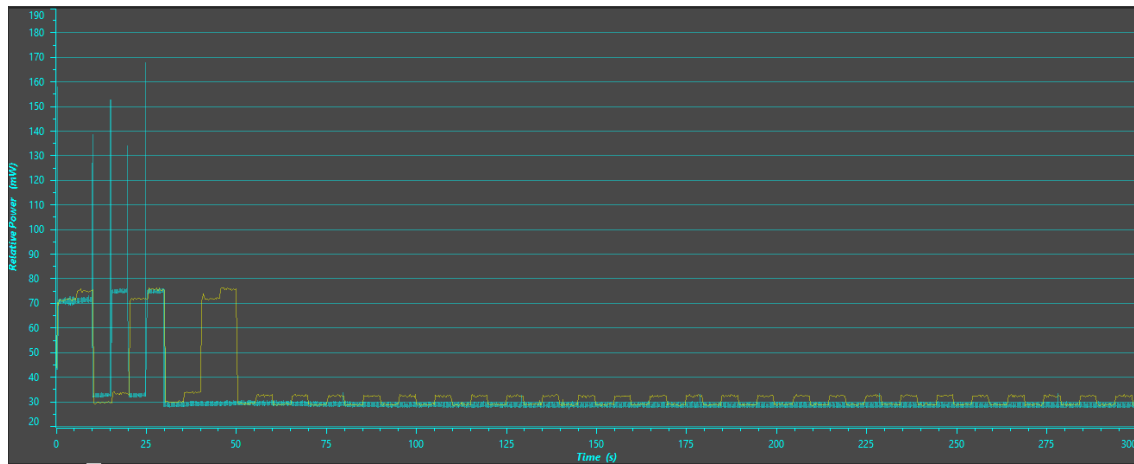


Figure 7 – Comparison between Hibernate Modes (Blue) and Low Battery (Yellow).

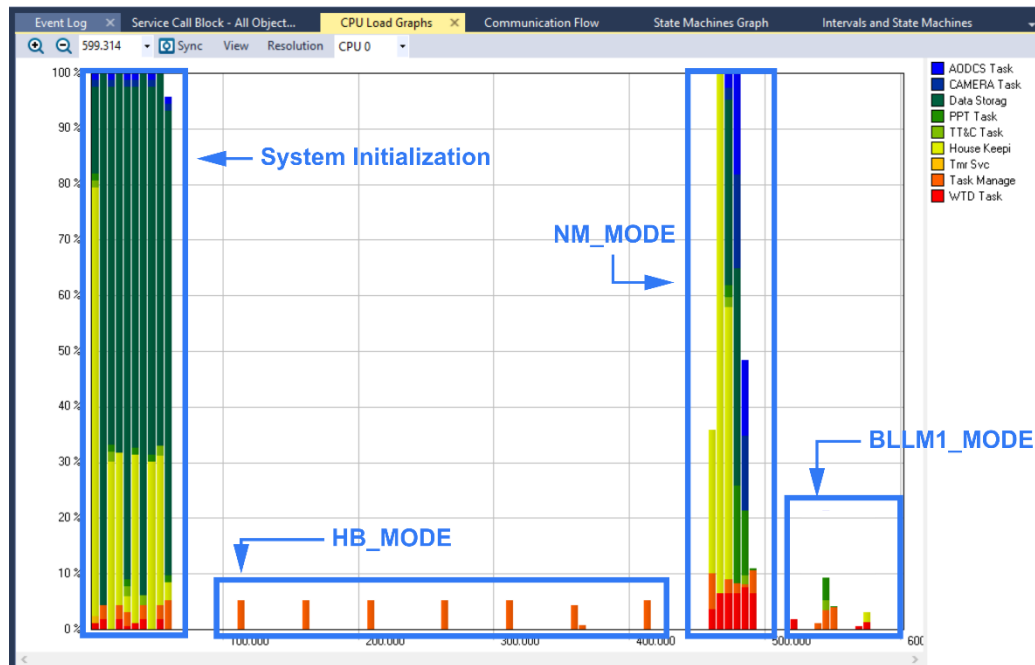
When comparing the performance of low power mode with hibernation mode, there was a 9% energy saving (0.6 days). This saving can be increased by changing the *SysTick* frequency, the FreeRTOS interrupter, to a lower frequency, e.g. 32KHz. The system in hibernation mode will need to be idle for a long time, so a lower *SysTick* frequency will be sufficient to perform kernel interrupts and will consume less than a 48MHz clock rate, as was used during the test. Figure 8 shows the comparison in graphical form.



**Figure 8** – Comparison between Hibernate Modes (Blue) and Low Battery (Yellow).

### 4.3. Operation Modes

To test the OBC state machine, it was used the software Tracealyzer from Percepio [13]. During the tests, it was possible to observe several interesting phenomena. At system startup, there is a high CPU usage, due to tasks, mutex, etc, memory allocation. When the light source was turned off, the system went into low power mode and only *TaskManager* was activated and running slower. The light source was turned on again and the system went into the nominal mode and all tasks run without CPU limit. After that, the brightness of the light source was decreased and the system was put into the low power mode, only the control (*WatchDogTask* and *TaskManager*) and data manipulation (*HouseKeeping* and *DataStorage*) tasks were performed.



**Figure 9** – Tracealyzer snapshot of the system running.

### 4.4. Anti-Lock System

To test Watchdog at software level, a system crash was simulated, causing a failure during the *TT&CTask* bit setting in the *WatchDogTask* handler. As expected, *WatchDogTask* checked the bits and deleted the *TT&CTask*, as shown in Figure 10. Before the lockup,



the *TT&CTask* had the label TT&C Task(2) [blue] and, after being restarted, changed to TT&CTask(3) [green].

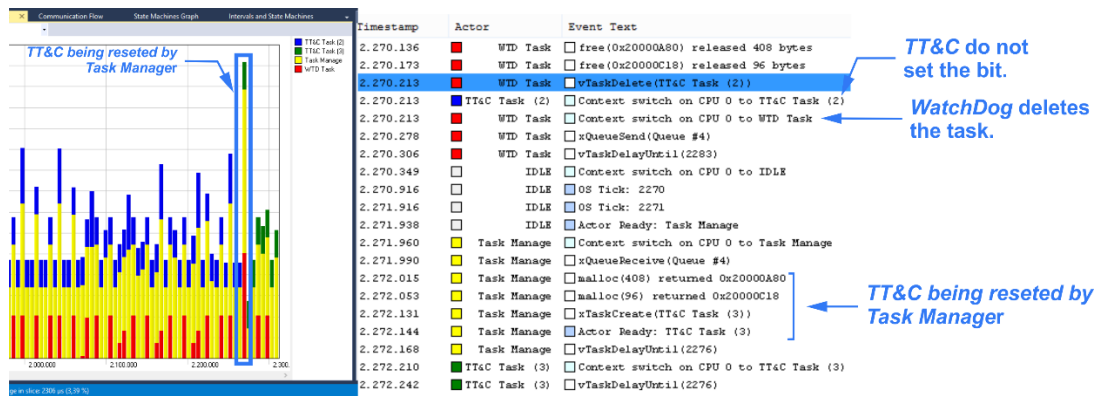


Figure 10 – TT&CTask lockup simulation.

#### 4.5. Printed Circuit Board

The PCB was designed using the KiCad EDA, an open-source software for circuitry design. The layout had two layers: a signal layer and a ground layer. In order to reduce cost and increase design flexibility, the chosen SMD components were 1206 package (approximately 3.2mm by 1.6mm). This choice allows components to be manually welded without the aid of robotic machinery. All traces were made using a 45° angle for every corner and were as short as possible to reduce inductance. The feed and ground connections were made using vias, saving space and decreasing the Electromagnetic Interference (EMI). It was not possible to have an intact ground plane and some traces were routed on the bottom layer, which may increase the EMI. The PCB dimensions are 95.89x90.17mm, which was based on the PC104 specification [14]. The final result is shown in Figure 11. The PCB manufacturing represents the current stage of the project. The next steps in developing OBC will be shown in section 6.

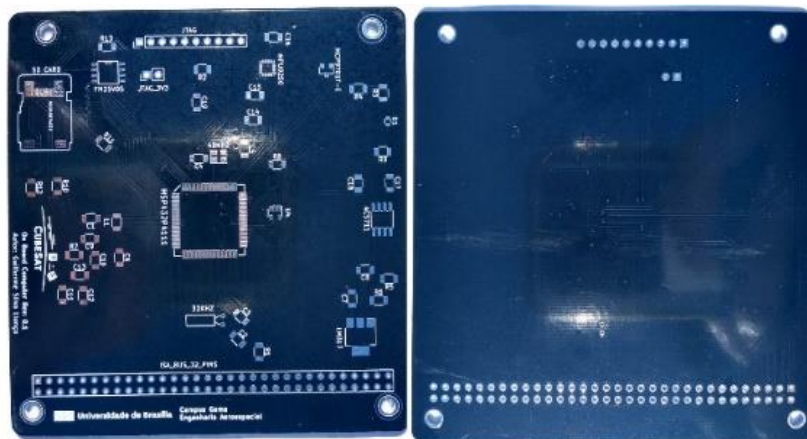


Figure 11 – PCB top and bottom view.

#### 5. CONCLUSION

This work presented the main advances made so far in the prototyping of UNB On Board Computer for the future CubeSat mission of the University of Brasilia. The development was divided into two parts: (a) Hardware, where the microcontroller and peripherals were selected; (b) Software, where the architecture and components for each layer were defined. According to the results, MSP432P4111 is a feasible choice for low power and

intermediate performance scenarios. Using FreeRTOS as a real-time operating system is also an adequate choice for low RAM microcontrollers. It was also seen that Watchdog at software level worked as a form of redundancy in cases of partial code lockup. In addition, preliminary software can now change state according to some input, for example brightness.

## 6. FUTURE WORK

Some established requirements have not been met due to the time available and the project complexity. All of these requirements will be resumed and implemented over the coming months to complete the first prototype of UnB OBC. A point of extreme priority to be resumed is the purchase and welding of the components, as it was not possible to test the PCB. After this step, the software tests may be performed in the OBC hardware, which means that the consumption and performance will be more accurate than in the TI-LaunchPad. The second point is the use of the 32KHz clock rate as the *SysTick* frequency source during satellite hibernation mode. It has been concluded that using a single clock for both high performances and hibernates mode does not make OBC robust in low battery scenarios. The use of multiple watchdog levels is not sufficient to decrease the risk of radiation effects on OBC. The use of COTS components decreases system reliability and other forms of protection should be considered.

## 7. REFERENCES

- [1] TEXAS INSTRUMENTS, MSP432P411x, MSP432P401x SimpleLink™ Mixed Signal Microcontrollers. p.214. (2018)
- [2] F. George, SwissCube HouseKeeping Parameters. p.1. Space Center EPFL, Switzerland (2009)
- [3] C. Frost and E. Agasid, Small Spacecraft Technology State of the Art. p.97. NASA Ames Research Center, California (2015)
- [4] ALLEGRO, High Sensitivity, 1 MHz, GMR-Based Current Sensor IC in SpaceSaving, Low Resistance QFN and SOIC-8 Packages. p.1. Manchester, New Hampshire (2018)
- [5] INVENSENSE, MPU-9250 Product Specification Revision 1.1. p.28. San Jose, California (2018)
- [6] MICROCHIP, Low-Power Linear Active Thermistor ICs. p.1. Chandler, Arizona (2016)
- [7] C. Frost and E. Agasid, Small Spacecraft Technology State of the Art. p.95. NASA Ames Research Center, California (2015)
- [8] TEXAS INSTRUMENTS, MSP432P4111 SimpleLink™ microcontroller LaunchPad™ development kit user's guide (Rev. B). p.11. (2019)
- [9] PC/104 Embedded Consortium. PC/104 Embedded Consortium. 2nd. p.25. (2008)
- [10] TEXAS INSTRUMENTS, USER'S GUIDE MSP432® Peripheral Driver Library. p.14. (2015)
- [11] R. Barry, Mastering the FreeRTOS STM Real Time Kernel: A Hands-On Tutorial Guide. p.17. Real Time Engineers Ltd., USA (2016)
- [12] B. P. Douglass, UML for the C programming language. p.12. IBM Corporation, USA (2009)
- [13] P. AB, Tracealyzer for FreeRTOS. p.1. Percepio AB, USA (2019)
- [14] PC/104 Embedded Consortium, PC/104 Specification – version 2. p.16, USA (2008)