UnB On-Board Computer Prototype for CubeSats

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

(1) Universidade de Brasilia
Faculdade Gama - Brasília, Brazil
Phone: +55 61 992278739, Mail: guilherme.lionzo@gmail.com
(2) Universidade de Brasilia
Faculdade Gama - Brasília, Brazil
Phone: +55 61 phone, Mail: santilli@aerospace.unb.br
(3) Universidade de Brasília
Faculdade Gama - Brasília, Brazil
Phone: +55 61 phone, Mail: aguayo@unb.br

Abstract: Researchers from the University of Brasília (UnB) are studying the feasibility of a CubeSat 3U mission, as a technology demonstrator. Some studies are already being carried out, in order to offer solutions for this future mission. The present research is aimed at the construction of an Onboard Computer (OBC) for this future mission. During the development of the OBC, it was used the co-design methodology, which allowed for the development of hardware and software at the same time. During the design of the theoretical project, it was chosen the microcontroller and another devices to compose the OBC's hardware. For the embedded software, the FreeRTOS operating system was defined as the 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.

1. INTRODUCTION

Immersed in a global process of capacity building and knowledge enhancement that achieves the ability and independence to perform small satellite missions, researchers at the University of Brasilia (UnB) are studying the feasibility of a 3U CubeSat satellite mission as a technology demonstrator. At UnB, some studies have been ongoing for several years to create and deliver solutions independently for these small future economies.

This paper aims to present the results of a research allowed so far and dedicated to the construction of an onboard computer (OBC) for this type of future mission.

2. OBC REQUIREMENTS

The requirements raised for the OBC are shown below.

- Control of a CMOS Camera (Complementary Metal-Oxide-Semiconductor);
- Control of a PPT (Pulsed Plasma Thruster);
- Control of an Inertial Sensor;
- Ensure a subsystem with a high level of confidence even without using radiation resistant devices;
- Nonvolatile data storage;
- Have anti-lock system;
- Change operating modes according to battery level.

3. OBC DEVELOPMENT

For the development of OBC, the Co-Design methodology was used meaning that the development of hardware and software happened simultaneously. In this section, we will show the components chosen for the hardware architecture and OBC software solution. Figure 1 shows the architecture of the OBC as well as the interfaces for communicating with the other subsystems.

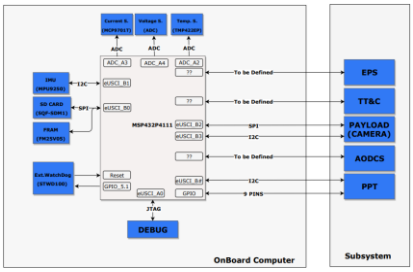

Figure 1 - OBC Architecture.

3.1. Hardware Architecture

Hardware development first began by selecting the microcontroller, as it is the component that will delimit OBC performance. The selection criteria considered: Low Consumption; ADC; GPIO; Serial; PWM; Clock; Temperature. From these criteria, we searched for microcontrollers from the most common manufacturers, such as Microchip and Texas Instruments. We opted for the selection of microcontrollers that had a development board, to enable the implementation time.

The selected microcontroller was the MSP432P4111 due to its intermediate performance and low consumption. In the following section there is a brief description of the microcontroller.

3.1.1. MSP432P4111

The MSP432P4111 has ARM Cortex-M4F as its microprocessor. This microprocessor features Reduced Instruction Set Computing (RISC) architecture with 32-bit instruction, and can operate at frequencies above 48MHz. It is designed for applications that require low power, high efficiency, good signal processing capability, low cost and easy usability. Such a microcontroller, known for its low power consumption, has 18 modes of operation and on average consumes 520uW / MHz. It has 4 16-bit timers, 24 14-bit ADC, 8 serial communication interfaces, real-time clock (RTC) block, and over 84 I / O pins. MSP432P4111 has 2048 KB Main Flash Memory; 32KB Flash Information Memory (Area for Bootloader, TVL and Flash MailBox); 256KB SRAM, including 8KB of backup memory. (Texas Instruments, 2018)

3.2. Software Architecture

To facilitate the development and abstraction of some embedded software interfaces, the layered architecture was used. This means that the user does not need to know very specific parts of the system, facilitating the usability and maintenance of the software. Below is the proposed software architecture for OBC.

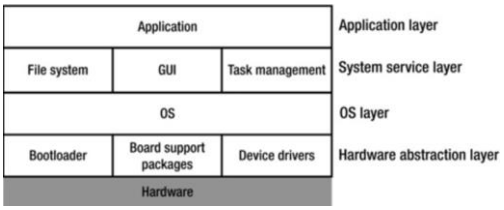

Figure 2 – Layered Abstraction Architecture.

Hardware Abstraction Layer

The HAL layer is played by the Driver Library (DriverLib) package, developed by Texas Instruments, which aims to facilitate the development of embedded projects and help code portability. Using this package, the developer does not need to know what happens at the registrar level, making development more user-friendly and faster (TEXAS INSTRUMENTS, 2018).

Operating system layer

The RTOS chosen for embedded software was FreeRTOS due to its wide use in CubeSat missions. This kernel, developed and maintained by Real Time Engineers Ltd, is freely distributed under the General Public License (GPL) license (BARRY, 2016). In the context of FreeRTOS, each running task is called a 'task'. In the context of the project, the use of tasks is critical to creating a certain level of abstraction and ensuring the requirement of Hard-RTOS.

System Service Layer

The System Service Layer (CSS) is the layer where mission-oriented routines are implemented. In the development of this layer, we used the FunctionalC standardization, extension of UML that allows the modeling of systems based on C language (Douglas, 2009). The State Machine and the Archives Diagram were elaborated.

The state machine contains four modes of operation: Nominal Mode, Safe Mode, Battery Low Level Mode, and Hibernate Mode, as shown in Figure 3 below. It is noteworthy that the pre-release stages were not taken into account and deployment.

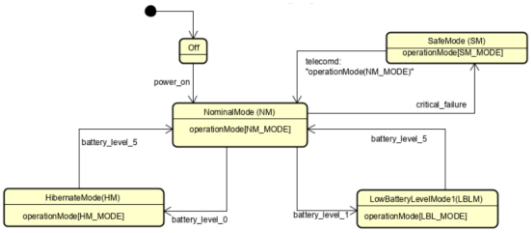

Figure 3 - Service Layer State Machine.

The File Diagram has 8 routines, as follows: one control (**TaskManager**); one for data collection (**HouseKeeping**); one of storage (**DataStorage**); a lock control (**WatchDogTask**); five referring to CubeSat subsites.

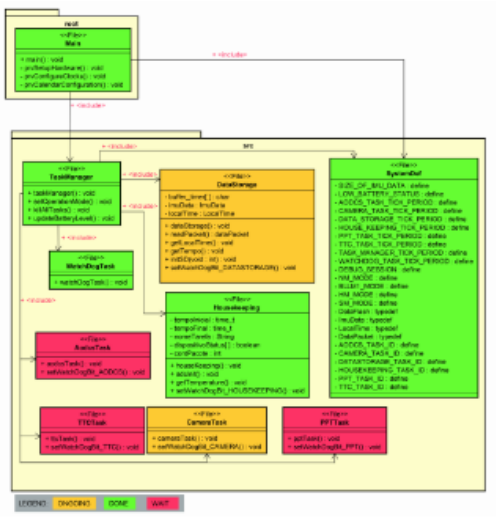

Figure 4 - System File Diagram.

4. RESULTS

Since OBC hardware was under construction, the embedded software was simulated on the MSP432 LaunchPad. A photoresistor was used to simulate the CubeSat EPS battery level. This approach is somewhat in line with reality because the incidence of light interferes with the amount of energy stored in the EPS. Figure 5 shows the connection of components in LaunchPad. In this photo there are the used COTS modules (SD Card, MPU9255 and photoresistor) and the camera (Paylod).

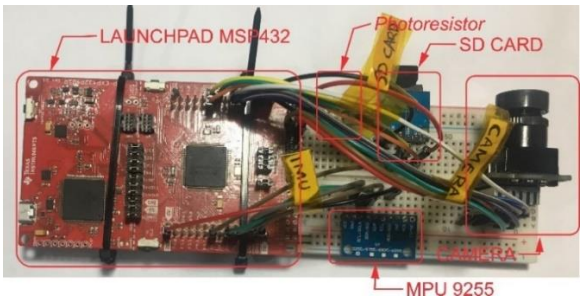

Figure 5 - Protoboard with COTS and LaunchPad components.

4.1. Data Acquisition and Storage

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

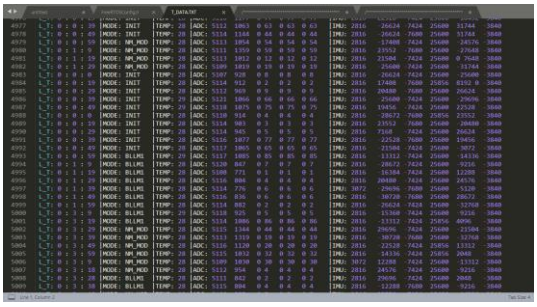

Figure 6 - Telemetry data stored on the memory card.

4.2. Power 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 Launchpad-powered modules / sensors. 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.

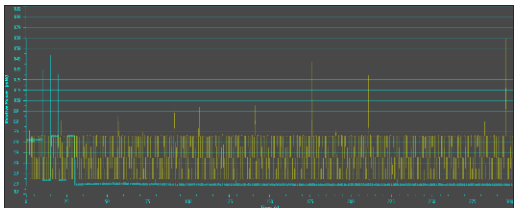

Figure 7 - Comparison between Hibernate Modes (blue) and Low Battery (yellow).

It was not possible to complete the same improvement by comparing low power mode with hibernation mode. There was only a 9% savings, 0.6 days. This is because hibernation is not fully optimized. The sync source during this state is still still 48MHz. Ideally, you should use 32KHz external clocking to perform kernel interrupts. Unfortunately due to the inexperience of the student with FreeRTOS, it was not possible to add a second sync source in low power mode. Figure 8 shows the comparison in graphical form, blue hibernation and nominal yellow.

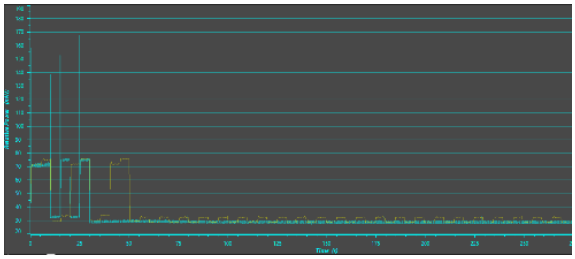

Figure 8 - Comparison between Hibernate Modes (blue) and Low Battery (yellow).

4.3. Operation Modes

To test the OBC state machine, the Percepio company Tracealyzer software was used. During the tests it was possible to observe several interesting phenomena. At system startup there is a high CPU usage, then the system goes into low power mode and only TaskManager is active and running slower. After hibernate mode, the system goes into nominal mode and all tasks run without CPU limit. And finally, the system was put into low-power mode, and only the control (**WTDTask** and **TaskManager**) and data manipulation (**HouseKeeeping** and **DataManager**) tasks were performed.

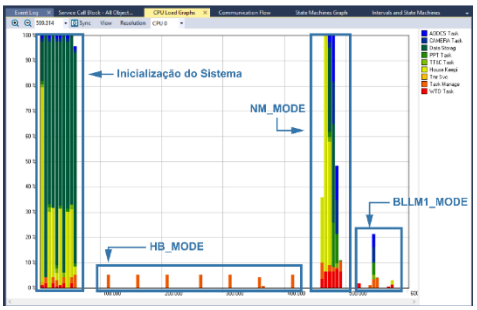

Figure 9 - System Tracealyzer snapshoot running in all modes.

4.4. Anti-Lock System

To test Watchdog at the software level, a lock on the TT&C Task was simulated, causing a bit not to be set in the **WatchDogTask** handler. As might be expected, the **WatchDogTask** checked the bits and deleted the TT&C Task, as shown in Figures 10 and 44. Prior to being flagged, the TT&C task had the label TT&C Task (2) [blue] and, upon restarting, changed to TT&C Task (3).) [green].

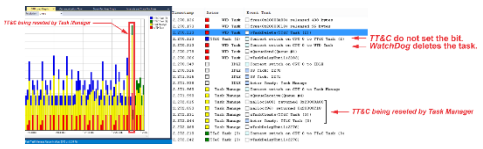

Figure 10 - TT&C Task Lock Simulation.

4.5. Printed Circuit Board

From the layout made in KiCad, we generated the *.gerbers* files that contain all the information for PCB manufacturing. The manufacturer PCBWay was used to manufacture the PCB. The end result is shown in Figure 11.

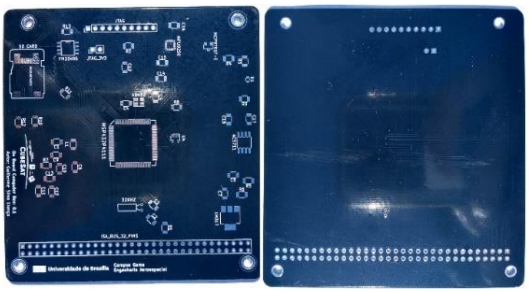

Figure 11 - Top and bottom view of the PCB.

PCB manufacturing represents the current stage of the project. The next steps in developing OBC will be shown in section 7.

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 initial part presented with a brief explanation of the requirements raised for the OBC. The development was divided into two parts: Hardware, where the microcontroller and peripherals were selected; Software, where the architecture and components for each layer were defined. It was concluded that using the TI-MSP432 is a great choice for low power and intermediate performance scenarios. Using FreeRTOS as a real-time operating system requires only little memory for systems, as well as using software-level watchdog. It was noted that the preliminary software can already change state according to some input, for example the brightness.

6. FUTURE WORKS

Some project requirements have not yet been met due to the complexity of the project and the limited time available. All of these points will be taken up, developed and further developed during the project to reach the conclusion of this UnB onboard computer prototype for CubeSats. An extremely important point to be taken up is the purchase and welding of the components, as it was not possible to test the board developed in the project. Another point to note is the use of the 32KHz clock as SysTick's timing source during satellite hibernation. It has been found that using a single clock for both high performance and hibernate 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

TEXAS INSTRUMENTS (a) . **MSP432P411x, MSP432P401x SimpleLinkTM MixedSignal Microcontrollers.** 2018. 214 p.