CONASAT-1 Cubesat: Integration of Environmental Data Collector

Alessandra U. Rodrigues(1), **Alysson P. Lima**(1), **Manoel J. Carvalho**(1),
Samaherni M. Dias(2), **José M. Duarte**(1), **Jefferson A. Silva**(1)
(1)National Institute for Space Research (INPE), Natal, RN, Brazil, email
(2)Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil

Keywords: *CubeSat mission, Flight software, Environmental data collection.*

This paper describes the integration of a payload into a nanosatellite. The project that is developing this integration has the goal to renew and continue the Brazilian Environmental Data Collection System, a system that needs environmental data from all the national territories for weather forecasting and climate studies. To achieve this goal, the system needs a platform for data collection, satellites that receive environmental data and send it back to the Earth, and ground stations that receive the data sent by the satellites. This article uses the CONASAT's project, developed by INPE-Natal, which has a CubeSat, class of a nanosatellite with a square-shaped, named CONASAT-1, and to process the environmental information that came from the ground stations, the satellite has a payload named Environmental Data Collector (EDC). To make the integration and functionality of the EDC with the platform was created a flight software based on C language with an operating system, FreeRTOS, which works through tasks for keeping the satellite working. For this was developed five tasks responsible for the supervision, monitoring, and data sending, and also four modes of operation that define all the actions necessary for the satellite operation, and the sending of the environmental data back to the ground stations.

1. Introduction

In the 90s, intending to provide the country with environmental data on the national territory, started the Brazilian Environmental Data Collection System (BDCS) with the launch of the satellite SCD-1 in 1993. Over the years the BDCS has grown with the launch of a few more satellites, but currently, most of these satellites are at the end of their useful life, which makes necessary new satellites for keeping the supply of data to monitor the territory [1].

Besides the satellites, ground stations and Data Collection Platforms (DCPs) also make up the BDCS [2]. The DCPs are small stations, usually installed at remote places, that send environmental data to the satellites, among the main data are meteorological (temperature, pressure, wind direction and speed, humidity)[3]. Those data are transmitted by the satellite and used in weather forecasts, ocean currents, tides, etc. Figure 1 below, shows a configuration of the BDCS with DCPs, ground stations, and a satellite.

Currently, the BDCS is compound for two ground stations (Cuiabá and Alcântara) for control and receive data, approximately 500 DCPs spread over several regions of the country and a few satellites, among them SCD1, SCD2, CBERS1, Floripasat, and Amazonia 1.

As the number of satellites in the BDCS is small and some of them are at the end of their useful life, the project CONASAT was created as an innovative solution

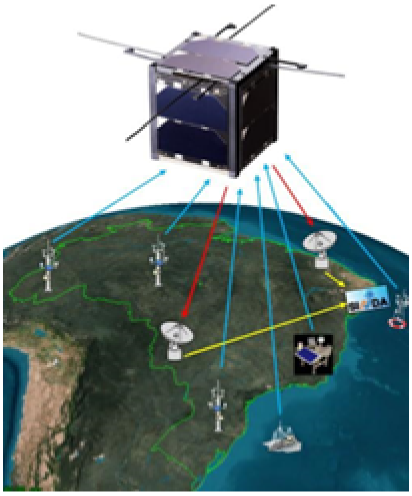


Figure 1: BDCS Illustration

for the space segment, based on a constellation of low-cost nanosatellites, to provide continuity to the BDCS[4].

The CONASAT-1, the first satellite of the constellation, is 1U CubeSat has a cubic shape, edges of 10 centimeters, and uses commercial off-the-shelf components (COTS) for its structure and electronics. The CubeSat design specification was created in the USA to provide a standard for the design of picosatellites to reduce cost and development time, increase accessibility to space, and sustain frequent launches[5]. The CONASAT has its hardware developed by the company EnduroSat and a payload developed by INPE'S (National Institute of Space Research) Northeast Space Coordination (COENE).

CubeSat can be targeted to various missions, an example is when they integrate a payload such as an image processing providing images for monitoring purposes of vast territories, borders and deforestation [6]. Another application of nanosatellites is an integration with the Global Navigation Satellite System (GNSS) which provides location and navigation but also studies the Earth's atmosphere, oceans, and land surface [7].

The CONASAT's payload is the Environment Data Collector (EDC), it is compatible with Cubesats platforms and was developed, as the satellite, as a low-cost and low-power solution to renew and expand the BDCS[8]. The EDC is a transponder of signals coming from the DCPs, with onboard decoding, capable of decoding up to 12 DCPs at the same time.

The focus of this paper is the integration and functioning of EDC with the CONASAT-1's platform, for this was developed a flight software responsible for the mission of the project. This software is based on a previous code made by EnduroSat, in which was added the entire interface communication with the EDC and a routine for its functioning, named EDC Mode. The goal of this mode is to monitor and supervise the satellite while the EDC is on, ensuring that the satellite keeps working properly and if something is not as it should take appropriate action.

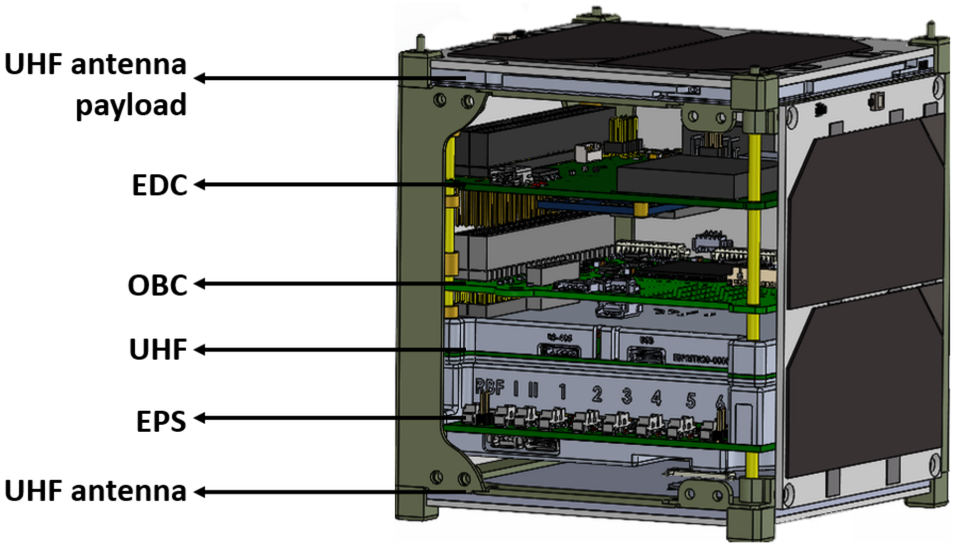


Figure 2: **Architecture of CONASAT-1**

2. CONASAT's Cubesat

The CONASAT-1 platform comprises the integration of antenna subsystem, power management subsystems (EPS), communications and command (UHF), data manipulation capabilities (OBC), attitude determination and control (ADCS), provided by EnduroSat, and the EDC as a payload, provided by INPE.

CONASAT-1's hardware is divided into subsystems, as in Figure 2. The communication with the satellite is established through an Ultra-High Frequency (UHF) half-duplex. This subsystem is a transceiver that uses a 2GFSK modulated signal with a typical baud rate of 9600 bps, a frequency offset of 2400 Hz, and a modulation index of 0.5. All of these communication features are controlled by EnduroSat's telemetry and telecommand protocol (ESTTC).

The entire platform is powered by the Electric Power System (EPS), which includes a battery with a capacity of 10.2 Wh and several voltage rails. Of this capacity, a typical OBC power consumption is approximately 0.2 W, for the EPS subsystem 0.08 W, UHF 1.25 W, and EDC 1.1 W.

The Onboard Computer (OBC) is the main control and data processing system on the platform, its functions are to control all other subsystems, collect and transfer data and perform attitude control-related calculations and commands. This subsystem is built on an ARM Cortex M4 microcontroller, which runs a FreeRTOS (CMSIS) instance as the basis for embedded software designed in a 4-layer structure, as shown in Figure 3. Also at the OBC runs the software of attitude determination and control, this subsystem is responsible for the satellite's orientation, it has sensors (solar sensors, magnetometers, gyroscope) to determine the satellite position and actuators (magnetorquer) to control it, this hardware it is spread over the satellite: solar sensor, gyroscope, and magnetorque are at the solar panels and the magnetorquer is on the OBC.

All embedded software was programmed in C language, and with the presence of the FreeRTOS scheduler, there are concurrent programming paradigms. The first layer presented in Figure 3 is dedicated to the software libraries for the microcontroller

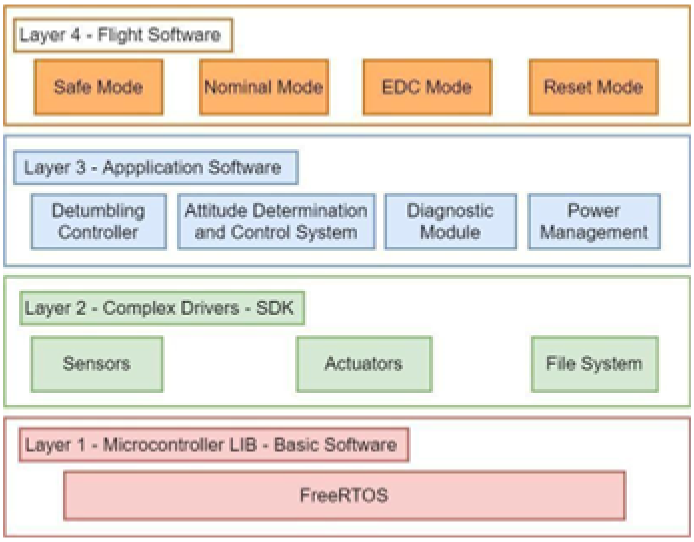


Figure 3: **Layers Structure of the embedded software**

peripherals, including the real-time operating system. The second layer includes the software drivers for the sensors — gyroscope, accelerometer, magnetometer, solar sensors, and actuators — magnetorquer. The third level of abstraction includes any application software for satellite stabilization and control, diagnostics, power management, etc. Finally, the focus of this work is the fourth level, which includes applications for satellite operation, integration with the payload, and manipulation of collected data.

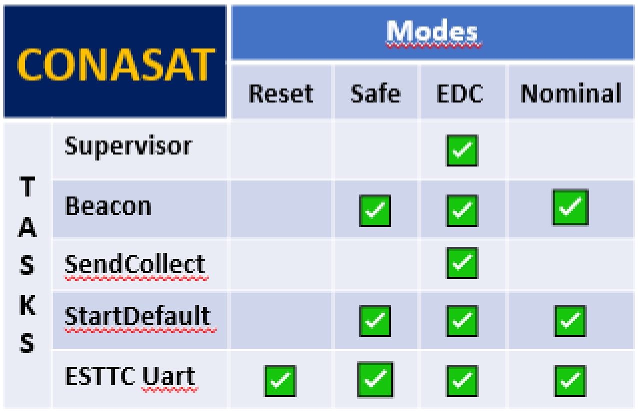
The upper layer, shown in Figure 3, is the Flight software developed for CONASAT-1, this software layer controls the satellite and sends requests to the lower layers. Flight Software includes several tasks that are running, those tasks include, but are not limited to:

- EDCColetEnv – Linked to the payload (EDC), activate the decoding of payload signals, request collected data, and send this data to ground stations;
- Beacon - To transmit a beacon to locate the satellite and transmit specific health data from the satellite;
- Supervisor – Connected to the payload (EDC), monitor data from the sensors to guarantee, in case of a contingency, protection actions;
- StartDefaultTask – Start sensors, print on both communication channels, and monitor task crashes;
- ESTTC_UART_TASK – Check and process new commands.

In addition to the tasks, the satellite also has several modes, where each mode is a combination of certain tasks, as shown in Figure 4. In addition, the satellite can transition from one mode to another by remote controls or under certain circumstances that can be defined. There are four implemented modes:

- Nominal: satellite only emits Beacon signaling. Maintenance telemetry can be collected and downloaded to ground stations when in view.

- Safe: satellite being kept secure (only critical systems turned on).
- EDC: operations with payload activation, data request from DCPs, data manipulation, and sending to ground stations when it flies over Brazilian territory.
- Reset: satellite prepared for reset (only the task - ESTTC_UART_TASK).



CONASAT		Modes			
		Reset	Safe	EDC	Nominal
TASKS	Supervisor			☑	
	Beacon		☑	☑	☑
	SendCollect			☑	
	StartDefault		☑	☑	☑
	ESTTC Uart	☑	☑	☑	☑

Figure 4: Modes and Tasks Overview

3. Integration of Environmental Data Collector

The EDC is physically integrated into the platform by a PC-104 connector, as are all the subsystems, and it is located above OBC. Is it responsible for receiving and decoding BDCS and Argos-2 signals in the 401.635 MHz ± 30 kHz frequency range. Its receiver is capable of decoding up to 12 signals simultaneously and storing up to 64 decoded messages in its memory. However, it is important to highlight that all the reception and decoding of signals performed by the EDC is done autonomously when a telecommand is processed by the OBC activating this subsystem.

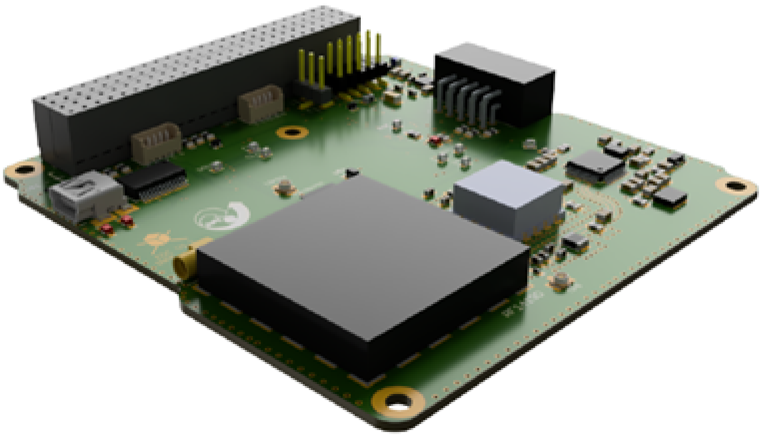


Figure 5: EDC

Furthermore, the EDC does not have a direct interface with the ground stations, it depends precisely on the OBC software, which is solely responsible for operating the EDC, requesting decoded data, handling and sending the ground stations via the UHF subsystem. The EDC has communication by I2C or RS-485, but in this project they are not used, all the communication between EDC and OBC is done by UART, as Figure 6, and using the driver development by the EDC team, some functions of this driver will be seen later, as EDC_Resume, EDC_PTT_POP, EDC_GetState, etc.

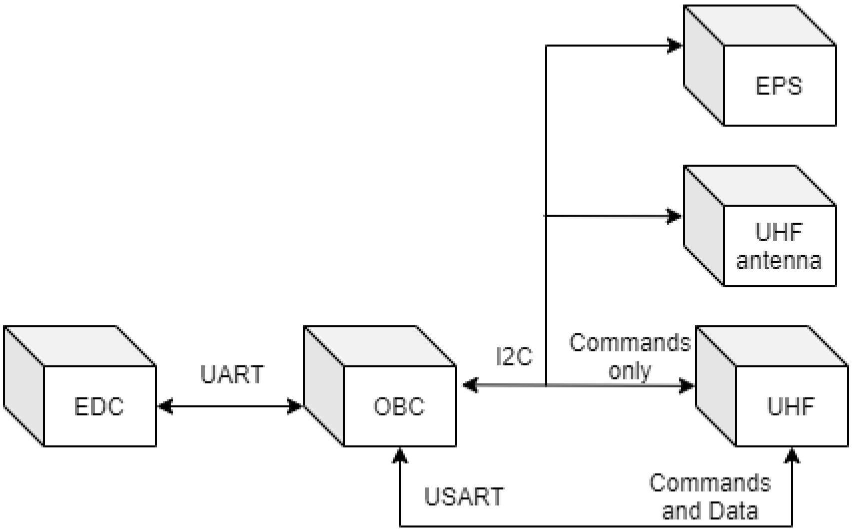


Figure 6: **Communication channels between subsystems**

The integration of CONASAT-1 with the EDC is fundamental to the BDCS's mission. To understand our contribution to this system it is necessary to detail the EDC mode. This mode is activated by telecommand from ground stations and governed by 4 concurrent tasks – EDCColetEnv, Supervisor, StartDefaultTask, and ESTTC_UART_TASK. The EDC's operation from its activation, by remote controls, to the transmission of its data, undergoes special characteristics. The EDC mode was programmed to operate only upon request from the stations on the ground and is automatically switched off by a 10-minute software timer after the start of the operation, in case of no change in the operating time.

3.1. Supervision task

The Supervision task aims to maintain the integrity of the EDC Hardware, by monitoring some electrical parameters provided by sensor readings present on the payload board. Among the parameters checked are voltage, currentSupplyD, currentSupplyA, and temperature. In summary, these parameters are obtained by the generation of housekeeping and compared individually with limits defined in the typical values, determined by the manufacturer, plus 25%. Whenever a violation of these limits is detected a counter variable is added and a new collection and verification is carried out, however when the failure counter reaches 5 detections, the EDC tasks are disabled, the subsystem is turned off and the Safe mode is activated.

3.2. StartDefaultTask

The StartDefaultTask task is responsible for ensuring the integrity of the software during EDC mode, ensuring that all tasks are running as planned. For this function, the independent watchdog (IWDG) was used to detect and resolve malfunctions due to software failures. It triggers a reset sequence when it is not updated within the expected time window, so it is up to the StartDefaultTask task periodically and with the lowest possible priority restarts the watchdog periodically. Therefore, in case the task scheduler error or any other task crashes the processing, the StartDefaultTask does not execute and the watchdog comes into action.

3.3. EDCColetEnv task

The main function of the payload is to receive signals from the Platform Transmitter Terminals (PTTs) belonging to the BDCS, for this, the EDC is composed of a Front End RF and a Processing Unit to facilitate its handling of internal data. The EDC can provide four types of data frames via telemetry: PTT Decoder Frame, HK Frame, System Status Frame, and ADC Sampler Frame, and OBC can select the output frame type at any time. Of these frames, the EDC Mode uses the first three, the PTT Frame is a package for every PTT signal received by EDC plus some order information such as frequency, power, and time; the HK Frame contains the most housekeeping information and the System Status Frame has information about the EDC currently status. The EDCColetEnv task, is an interaction routine between OBC and EDC focused on requesting PTT Decoder Frames, processing, and sending them to ground stations. Figure 7 illustrates its operation: when the satellite receives the command to turn on the EDC, is started the decoding of the PTTs by EDC PTT_Resume, next the OBC request to the EDC the HK Frame (EDC Housekeeping) and the state of the EDC (EDC Get_State), from the frame state is verified if the decoding is working, in the case it is not working, a PTT_Resume is done again, otherwise, the OBC asks the EDC a PTT Frame, in this step is verify the amount of PTT storage at the EDC and, existing a number higher than zero (state.pttAv), one by one they are sent to OBC and storage, to free space at the EDC is performed EDC PTT_POP, responsible for removing the oldest PTT package from the buffer in the EDC. This step of picking up the PTTs from EDC is done while the EDC has packages, until the next EDC Get_State. When there are no more PTTs, the OBC will mount a large package with all the PTTs storage and the housekeeping frame already received, to this package will be added to header information and then the package will be sent in several small packages of 128 bytes.

4. Integration Tests

To test, a test bench was set up with three sections: PCDs Emulator, Ground Station, and CONASAT Satellite. In Figure 8, it is possible to see that the first section is formed by a computer and a radio Universal Software Radio Peripheral (USRP), that is responsible for emulating the PCDs, this is made by simulating DCPs at the computer using the software GNU Radio, the USRP send the data emulated through RF antenna for the 401.635 MHz antenna at the satellite. The PCD data arrives at the satellite, is decoded by EDC, bundled in packages by EDCColetEnv, and sent back to the Earth through the UHF antenna at 462 MHz, which leads to the third section, Ground Station, that represents a real Ground Station, but in this case, is composed by an antenna and a UHF transceiver just like the one at the CONASAT's platform, for this setup, the ground station sends a telecommand and receive telemetry using the software ScripCommunicator.

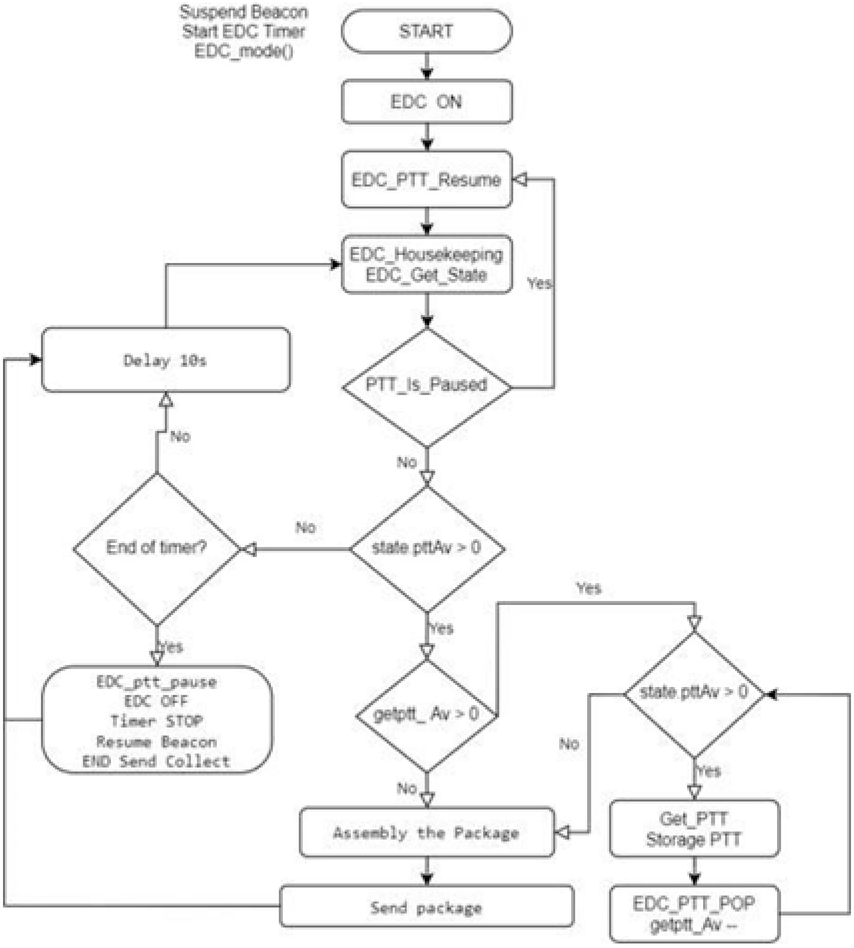


Figure 7: EDCColetEnv flowchart

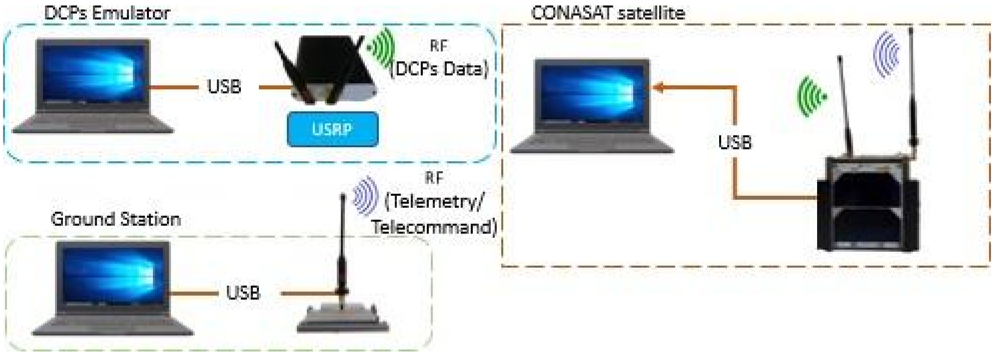
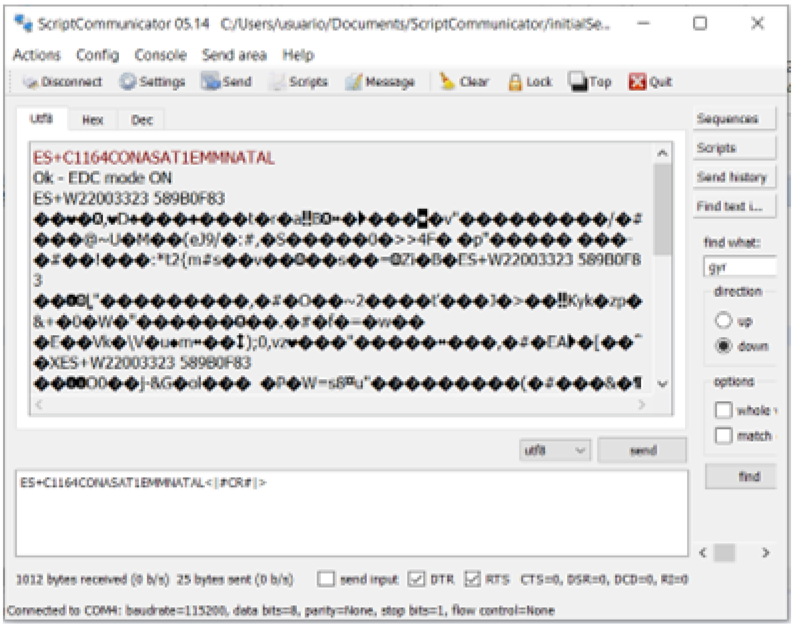


Figure 8: Test setup

4.1. EDCColetEnv test

This setup is an approximation of the BDCS made in a room of approximately 9m² in an area with 1.5 meters of distance between each section. In this room, a few tests were made to prove that the software was working as it should. The first one was in the case of normal operation, CONASAT-1 flies over Brazilian territory initially in Nominal Mode, then ground operations stations send the telecommand activating EDC mode as described in the following Figure 9, the satellite responds with a message the activation of the mode “Ok - EDC mode ON” followed by the telemetry of EDCColetEnv, the part of the message “ES+W22003323 589B0F83” it is an internal command of UHF transceiver to let it a message pass through the UHF and don’t make a part of the telemetry.



```
ES+C1164CONASAT1EMMNATAL
Ok - EDC mode ON
ES+W22003323 589B0F83
```

```
ES+C1164CONASAT1EMMNATAL<|#CR#|>
```

```
1012 bytes received (0 b/s)  25 bytes sent (0 b/s)  CTS=0, DSR=0, DCD=0, RI=0
Connected to COM4: baudrate=115200, data bits=8, parity=None, stop bits=1, flow control=None
```

Figure 9: **Reception of data collected**

To check the quality of the shipment, a bit flipping was done, in 1 byte of the package, on the frames from de EDC, choose randomly. Each package has a maximum of 49 bytes, and the last of them is the checksum, which is a way to verify the frame, so the test is all about checking the checksum. The tests were made with 1000 packages and three types of tests were done, the results will be shown in the table below:

- Without bit flipping;
- With bit flipping on all packages;
- With bit flipping on 10% of the packages.

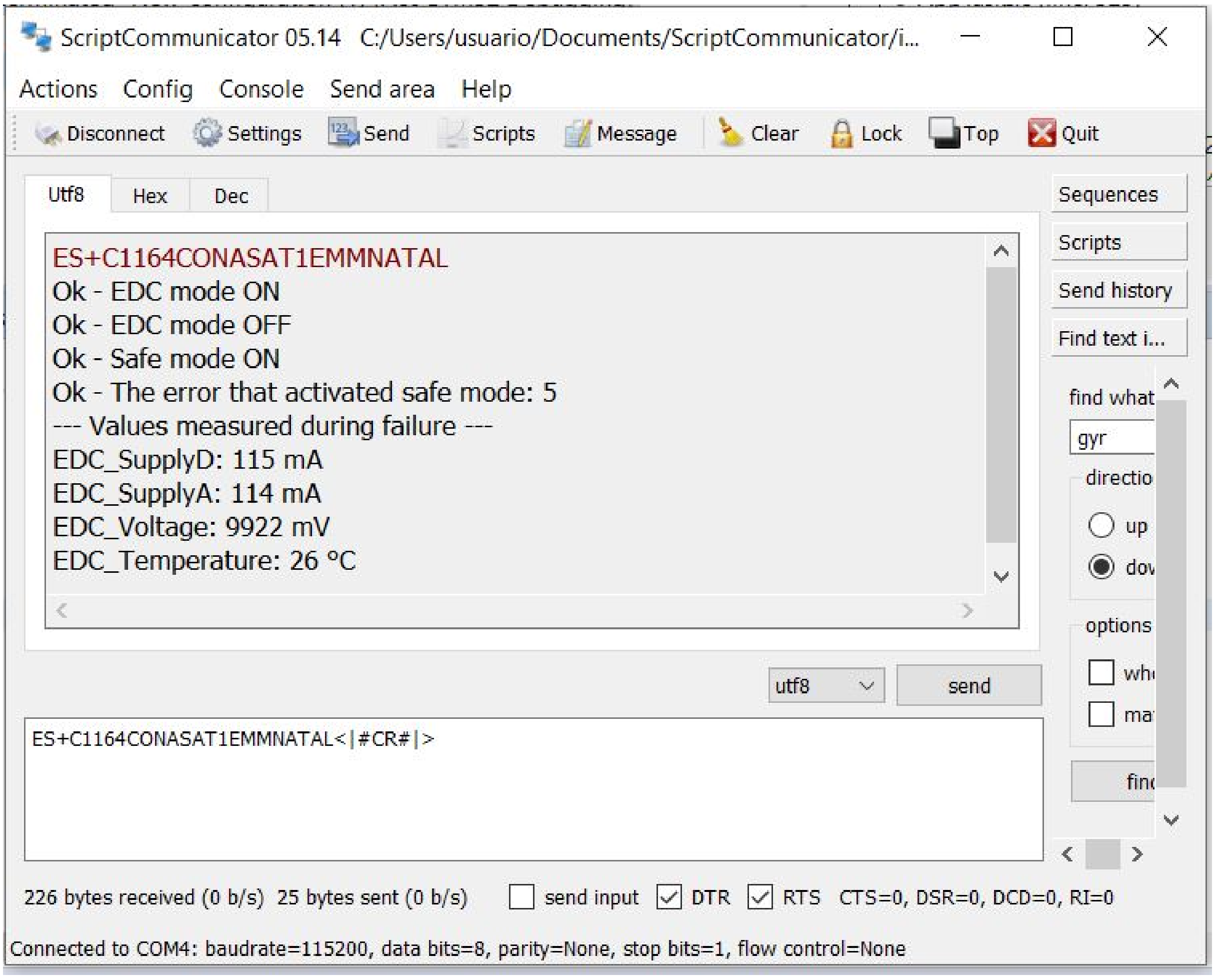
Table 1: **Error identification.**

Number of packages	Number of bit flipings	Identification of error(%)
1000	0	0
	1000	100
	100	10

The table above shows that the software development, for the test’s conditions (small distances and interferences), always gets right, regardless of the number of bit flipping, even in the case of inject error, the software always identifies the correct quantity of injected error, be in every package or in a few.

4.2. Supervisor task

In the same mode, for testing the StartDefaultTask task, the EDC subsystem is turned on and starts decoding the PTT signals, later the OBC requests this data releasing the EDC to collect and store new PTTs, processes and sends them to ground stations. With correct functioning, these data are requested and transmitted every 10 seconds to stations on the ground, which in turn are responsible for processing and inserting this data into SINDA. The Supervisor task periodically requests the electrical parameters of the EDC via sensors, so as soon as a violation of the tolerance limits is detected, the subsystem is turned off and its tasks are disabled, and Safe Mode is activated. The Figure 10 depicts the detection of an over voltage (listed as error 5) of 9922 mV, where the threshold is 6250 mV. To test this task, failures was simulated by increasing the value of the measurement made by the EDC's sensor measure.



```
ES+C1164CONASAT1EMMNATAL
Ok - EDC mode ON
Ok - EDC mode OFF
Ok - Safe mode ON
Ok - The error that activated safe mode: 5
--- Values measured during failure ---
EDC_SupplyD: 115 mA
EDC_SupplyA: 114 mA
EDC_Voltage: 9922 mV
EDC_Temperature: 26 °C
```

```
ES+C1164CONASAT1EMMNATAL<|#CR#|>
```

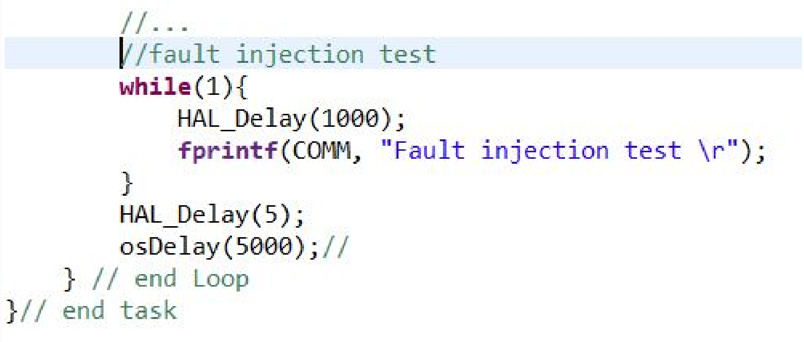
226 bytes received (0 b/s) 25 bytes sent (0 b/s) ☐ send input ☑ DTR ☑ RTS CTS=0, DSR=0, DCD=0, RI=0

Connected to COM4: baudrate=115200, data bits=8, parity=None, stop bits=1, flow control=None

Figure 10: Supervisor Task - Result

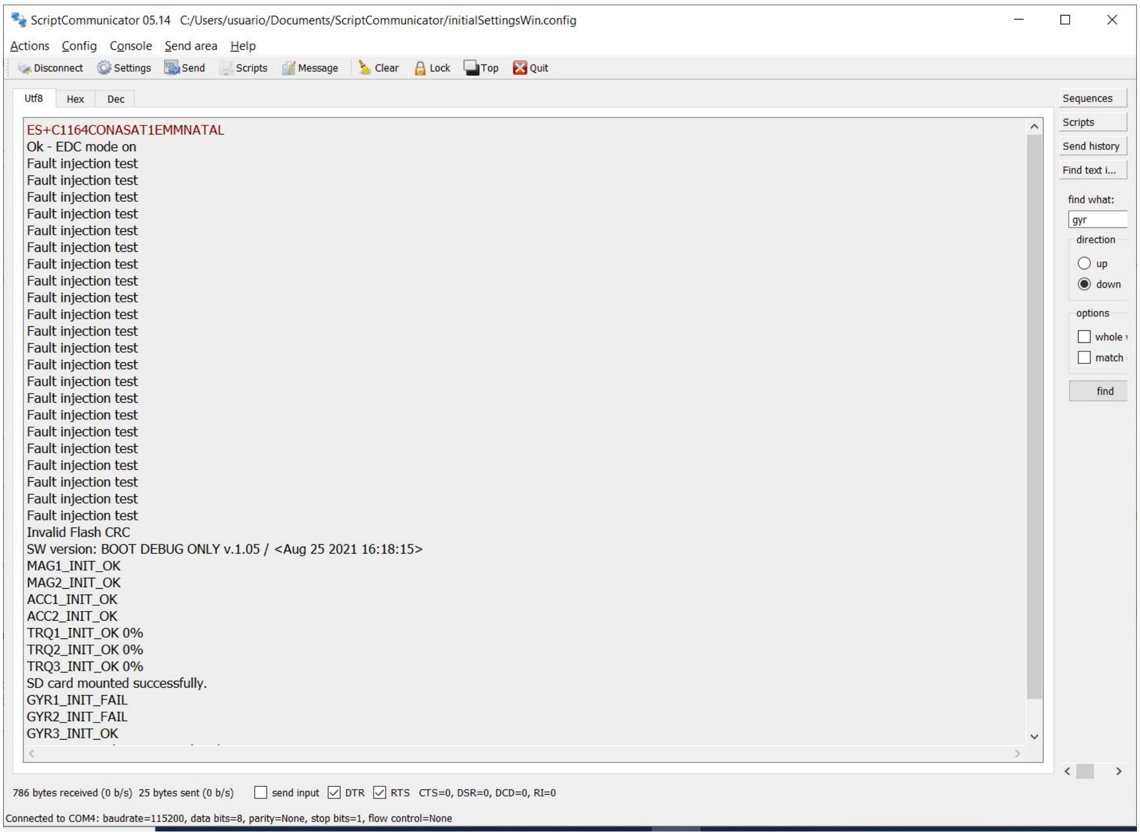
4.3. StartDefaultTask test

For the task integrity tests, flow blocking points were inserted in order to simulate the processing crash within the tasks. One of these points is depicted in Figure 11, inserted into the EDCColetEnv task before a context switch. The fact is that as the StartDefaultTask task has the lowest execution priority, consequently these locks prevent its execution and the Watchdog refresh results in its action as shown in the following Figure 12. So if there is a lock on running tasks, platform integrity is guaranteed with system reboot. During reset errors are computed, the EDC subsystem is done and tasks are disabled. Naturally, a platform reverting to Nominal Mode leads to receiving new remotes. The disadvantage of this technique is that despite avoiding blocking the platform, it is not possible to identify in which task the failure occurred. Reset is signaled for ground stations as shown in the Figure 12.



```
        //...
        //fault injection test
        while(1){
            HAL_Delay(1000);
            fprintf(COMM, "Fault injection test \r");
        }
        HAL_Delay(5);
        osDelay(5000);//
    } // end Loop
}// end task
```

Figure 11: Fault injection test - StartDefaultTask



```
ES+C1164CONASAT1EMMNATAL
Ok - EDC mode on
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Fault injection test
Invalid Flash CRC
SW version: BOOT DEBUG ONLY v.1.05 / <Aug 25 2021 16:18:15>
MAG1_INIT_OK
MAG2_INIT_OK
ACC1_INIT_OK
ACC2_INIT_OK
TRQ1_INIT_OK 0%
TRQ2_INIT_OK 0%
TRQ3_INIT_OK 0%
SD card mounted successfully.
GYR1_INIT_FAIL
GYR2_INIT_FAIL
GYR3_INIT_OK
```

786 bytes received (0 b/s) 25 bytes sent (0 b/s) ☐ send input ☑ DTR ☑ RTS CTS=0, DSR=0, DCD=0, RI=0

Connected to COM4: baudrate=115200, data bits=8, parity=None, stop bits=1, flow control=None

Figure 12: Fault injection test - StartDefaultTask - Result

5. Conclusion

In this article, the first version of the flight software developed for the CONASAT-1 was implemented with applications on operating modes, focusing on the integration of the platform with its payload (EDC), environmental data were received, treated, and sent to a ground station by telecommand and telemetry. Furthermore, tasks were carried out to ensure the integrity and proper functioning of the set of Hardware and Software.

The test setup intended that the design and development of the flight software meet the requirements of the platform and the mission application.

References

[1] K. P. Queiroz, S. M. Dias, J. M. Duarte, M. M. Carvalho, Uma solução para o sistema brasileiro de coleta de dados ambientais baseada em nanossatélites, HOLOS 7 (2018) 132–142.

[2] W. Y., V. O and S. P., Sistema Brasileiro de Coleta de Dados Ambientais: Status e planos futuros, in: Proceedings of the XIV Simpósio Brasileiro de Sensoriamento Remoto, Natal, RN, pp. 1633–1640. April 25-30.

[3] Instituto Nacional de Pesquisas Espaciais,INPE, Sistema nacional de dados ambientais, 2021.

[4] M. C, The New Brazilian Environmental Data Collection System, in: United Nations/South Africa Symposium on Basic Space Technology, Stellenbosch, South Africa, pp. 00–00. April 15-17.

[5] Cal Poly, Cubesat design specification cubesat design specification rev. 13 the cubesat program, 2014.

[6] D. Cahyono, A. Nugroho., Design and realization camera controller for a remote sensing payload of nanosatellite fpga (field programmable gate array) system based, in: 6th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, pp. 00–00. April 12-14.

[7] Li Qiao; E. Glennon; A. G. Dempster; S. Chaoui. , Using cubesats as platforms for remote sensing with satellite navigation signals, in: IEEE International Geoscience and Remote Sensing Symposium, IGARS, pp. 00–00. 2013.

[8] J.D. , CubeSat Payload for Environment Data Collection, in: United Nations/Brazil Symposium on Basic Space Technology, Natal, RN, pp. 00–00.