



HAL
open science

Reconfiguration of microcontroller payload driver

Romain Boyer, Frédéric Camps

► **To cite this version:**

Romain Boyer, Frédéric Camps. Reconfiguration of microcontroller payload driver. Cusat Symposium 2023, Dec 2023, Louvain, Belgium. 2023. hal-04330627

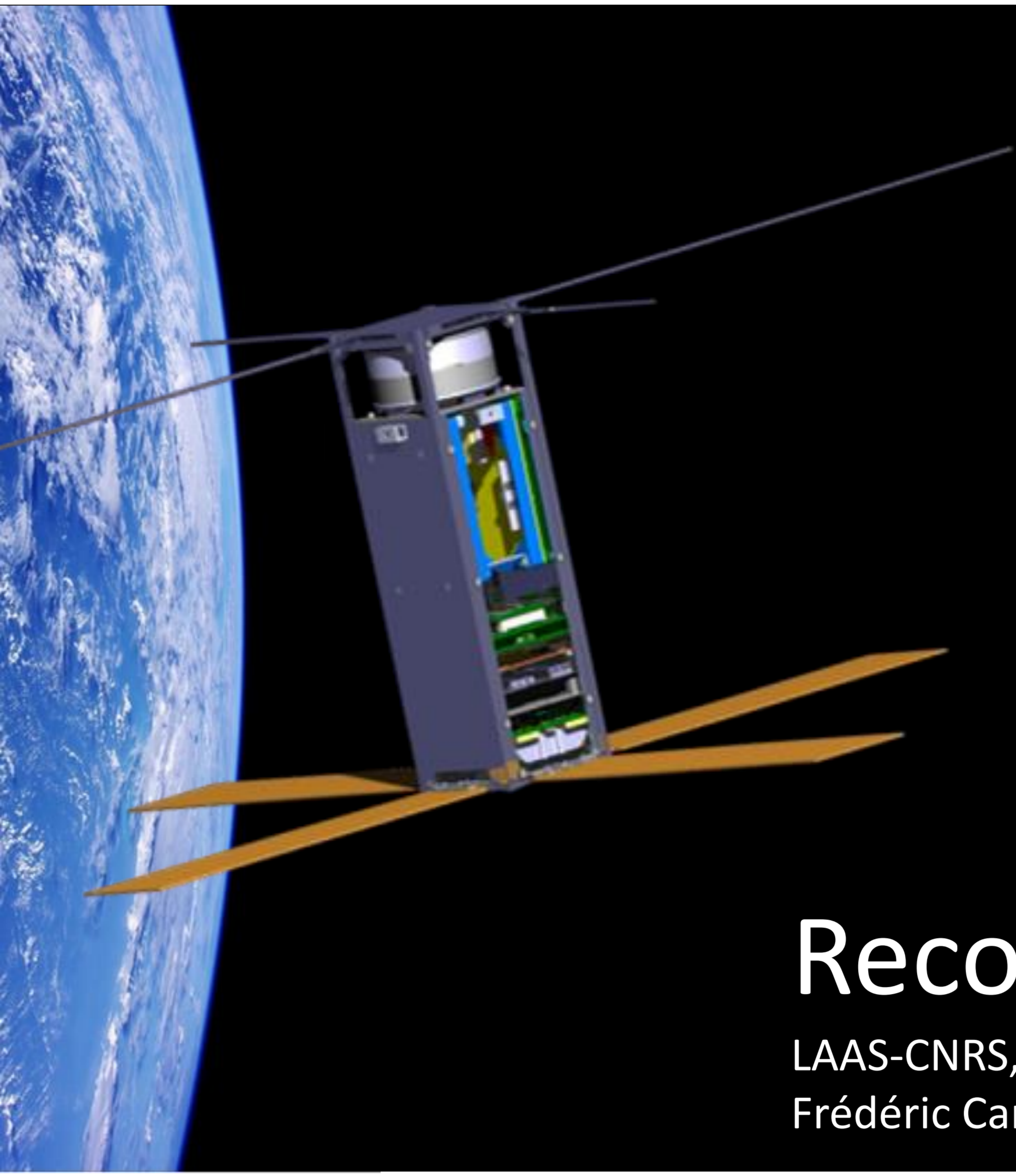
HAL Id: hal-04330627

<https://hal.science/hal-04330627>

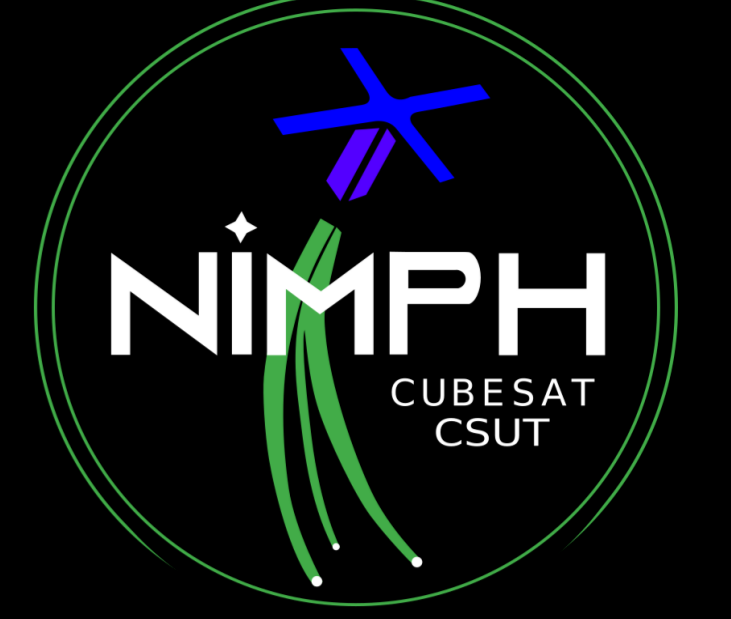
Submitted on 8 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



NIMPH, Nanosatellite to Investigate Microwave Photonics Hardware



NIMPH summary

The project:

NIMPH is an academic research project led by the University Space Centre of Toulouse (CSUT). It involves numerous Master's students from Paul Sabatier University (UPS) and the ENSEIHT engineering school, mainly through internships, research projects and practical work, under the supervision of professors and permanent staff from LAAS - CNRS.

Nanosatellite characteristics :

- Volume 3U (300 cm³)
- Weight 2 kg
- UHF radio (430 MHz)
- Sun-synchronous low-Earth orbit
- Power consumption: 7 W
- Mission duration: 2 years
- 3 payloads: EDMon, RADMon, M2M
- C phase

EDMon (NIMPH Payload):

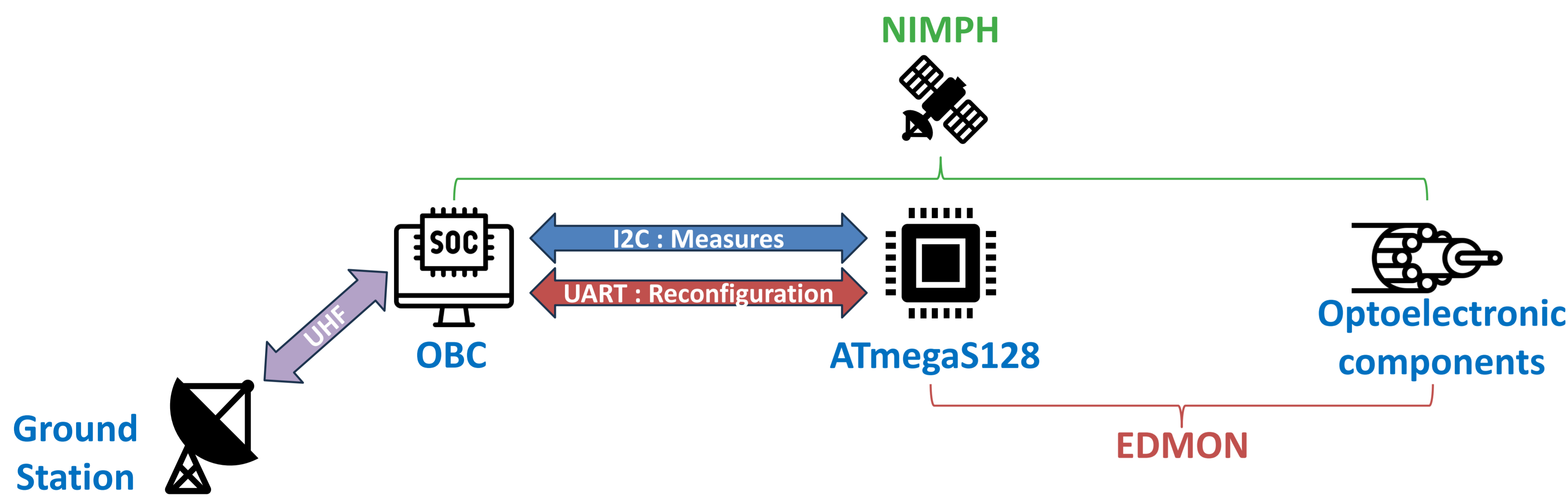
The NIMPH project's EDMon payload is designed to evaluate the influence of solar radiation and more generally space environment on the performance (gain & noise figure) of three Erbium doped fiber optic amplifiers (EDFA). The main challenges are the measurement reliability and the miniaturization of the payload. A microcontroller ATmegaS128 drives electronics components.

Reconfiguration of microcontroller payload driver

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

Frédéric Camps, Romain Boyer, Cyril Campanella / Contact fcamps@laas.fr, rboyer@laas.fr

Objectives and System Architecture:



Context and Objectives :

- Embedded systems in space are subjected to various phenomena that may **disrupt system operations** and damage memory components, such as radiation, wear and tear of electronic components, and degradation of electronic cards due to severe temperature cycling. Therefore, there is a **need to reprogram** a nanosatellite

payload driver to enhance its resilience against degradation.

- To mitigate potential memory damage and prevent program failure, the ATmegaS128 must include a specific program to modify the FLASH memory (128 kB) and reload a new mission software (MSW). The main objective is to **make the EDMON payload reconfigurable via UART**

communication. The program that will read and write in its own memory will be called **"bootloader"**.

- In order to enhance **system resilience** and address potential failure modes within the **failure tree**, such as electronic, optical, or mission **reconfiguration issues**, the bootloader is also used as part of the objective and system architecture.

Architecture :

- **Payload EDMon NIMPH:** Major payload of the NIMPH nanosatellite with optoelectronic components. To control them, an ATmegaS128 microcontroller is integrated into one of the two electronic cards of the payload.
- **On-Board Computer (OBC):** The on-board computer is a Soc Z7030 ("**Ninano**") electronic card developed by CNES and STEEL. It is responsible for controlling every single part of the satellite, including the microcontroller from the EDMon payload thanks to a spatial hypervisor XNG (FentISS). An I2C serial bus is used between them to transmit data and issue commands (set DAC, get ADC...). A UART serial bus is used to reconfigure the EDMon microcontroller (transmit binary programs, reconfiguration orders...).
- **Ground Station SCC:** A ground station SCC from U-Space / CNES is used to send Telecommands (TC) for reconfiguration that includes EDMon flight programs within the data fields. These Telecommands are received by the OBC before starting a reconfiguration of the EDMon controller.

Memory Architecture:

FLASH Memory Architecture:

- ATmegaS128's FLASH memory (128kB) is divided in two sections: **Application section** (blue, 120kB) and **Bootloader section** (orange, 8kB).
- Application Section hold **main program**, so then flight measurement program
- Bootloader Section hold **bootloader program** which is executed after a hardware reset.

Application Section Subdivision:

- Application section is divided in **six slots of 20kB**.
- The first slot is in the **running slot** (where the current running program is located). Due to interrupt table location, only this slot can be run.
- Slot 1 to 3 are **backup slots** where backup programs are stored.
- Slot 4 is the **security slot** where the flight mission software program is stored. If an error occurred on UART bus, this slot is restored in running slot number 0.
- Finally, slot 5 is a **configuration slot** which keep the actual configuration of memory as number of pages in each slot, ID and software versioning, bootloader versioning ...

Address :		Size :
0x00000	Slot 0	20kB
0x04FFF 0x05000	Running Slot	20kB
0x09FFF 0x0A000	Slot 1	20kB
	Backup Slot	20kB
0x0EFFF 0x0F000	Slot 2	20kB
	Backup Slot	20kB
0x13FFF 0x14000	Slot 3	20kB
	Backup Slot	20kB
0x18FFF 0x19000	Slot 4	20kB
	Launch MSW Slot	20kB
0x1DFFF 0x1E000	Slot 5	20kB
	Configuration Slot	20kB
0x1FFFF	Bootloader program	8kB

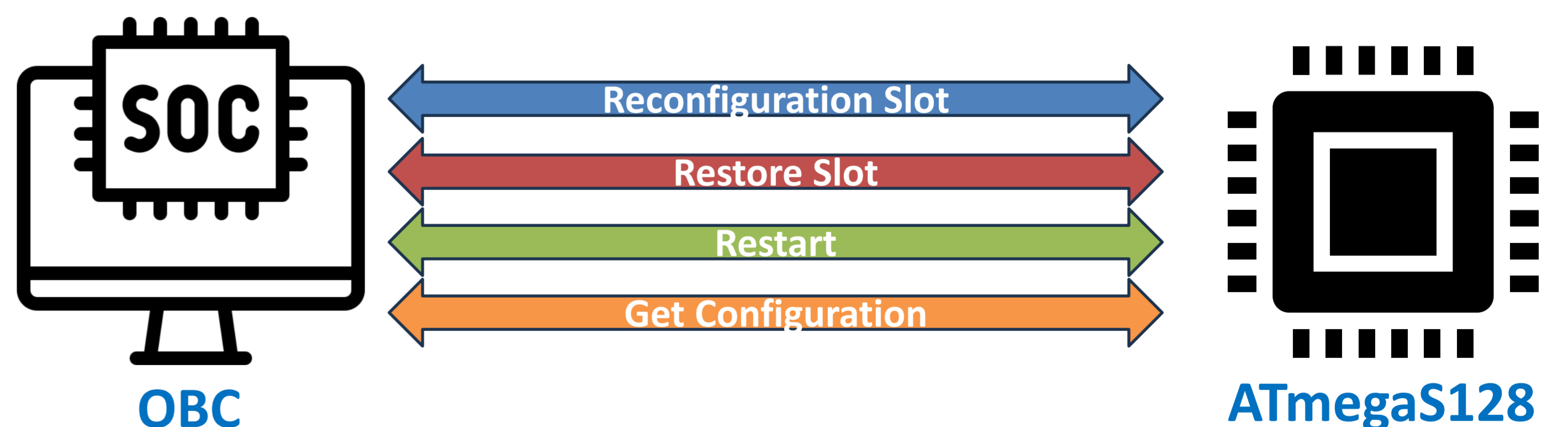
Legend :	
	Application Section
	Bootloader Section

Communication protocol:

Reconfiguration Commands:

On-Board Computer has four commands available in order to proceed to a reconfiguration function. Therefore, these commands are integrated in the bootloader program:

- **Reconfiguration Slot:** Ask for write a program in a specific slot number. This slot number is asked by ATmegaS128 before data transfer.
- **Restore Slot:** Ask for copy a specific slot number into the running slot (slot number 0).
- **Restart:** Ask for a restart of the running program. This command simply exit bootloader program and jump to application section (address 0x00000).
- **Get configuration:** Ask for the information in the configuration slot.



Communication protocol and Safety:

Communication protocol is essential to ensure resiliency in reconfiguration. Here are some examples of communication messages between OBC and ATmegaS128.

- Robust design of this UART system is essential for the mission and durability of the nanosatellite. Effective management of Hamming distance ensures increased resiliency to space radiation, thereby ensuring the success of this space mission.
- To secure heavy data transfer, cyclic redundancy code (CRC) is applied to data. The algorithm used is **CRC16-CCITT FALSE** (ESA ECSS-E-ST-70-41C, Annex B.1). The same CRC is used in PUS protocol (Ground Station – NIMPH communications). This ensure resiliency of the written bootloader.