Received: 4 August 2020 | Revised: 6 November 2020 | Accepted: 6 December 2020

DOI: 10.1002/spe.2947

EXPERIENCE REPORT

WILEY

Control and data acquisition software of the high-energy particle detector on board the China Seismo-Electromagnetic Satellite space mission

Alessandro Sotgiu[1] | Cinzia De Donato[1] | Claudio Fornaro[1,2] | Sandro Tassa[3] | Marco Scannavini[3] | Dario Iannaccio[4] | Giovanni Ambrosi[5] | Simona Bartocci[1] | Laurent Basara[6] | Roberto Battiston[6,7] | William J. Burger[6] | Donatella Campana[8] | Luca Carfora[1,9] | Guido Castellini[10] | Piero Cipollone[1] | Livio Conti[1,2] | Andrea Contin[11,12] | Fulvio De Persio[1] | Cristian De Santis[1] | Francesco M. Follega[6,7] | Cristina Guandalini[12] | Maria Ionica[5] | Roberto Iuppa[6,7] | Giuliano Laurenti[12] | Ignazio Lazzizzera[6,7] | Mauro Lolli[12] | Christian Manea[6] | Matteo Martucci[1,9] | Giuseppe Masciantonio[1] | Matteo Mergé[1,13] | Giuseppe Osteria[8] | Lorenzo Pacini[10] | Francesco Palma[1,13] | Federico Palmonari[11,12] | Beatrice Panico[8] | Alexandra Parmentier[1] | Francesco Perfetto[8] | Piergiorgio Picozza[1,9] | Mirko Piersanti[1] | Michele Pozzato[12] | Matteo Puel[6] | Irina Rashevskaya[6] | Ester Ricci[6,7] | Marco Ricci[14] | Sergio Bruno Ricciarini[10] | Valentina Scotti[8,15] | Roberta Sparvoli[1,9] | Bruno Spataro[14] | Vincenzo Vitale[1] | Simona Zoffoli[16] | Paolo Zuccon[6,7]

[1]INFN, Sezione di Roma “Tor Vergata”, Rome, Italy

[2]Uninettuno University, Rome, Italy

[3]Arakne s.r.l., Rome, Italy

[4]Innodesi s.r.l., Rome, Italy

[5]INFN, Sezione di Perugia, Perugia, Italy

[6]INFN - TIFPA, Povo (Trento), Italy

[7]University of Trento, Povo (Trento), Italy

[8]INFN - Sezione di Napoli, Naples, Italy

[9]Department of Physics, University of Rome “Tor Vergata”, Rome, Italy

[10]IFAC-CNR, Sesto Fiorentino (Florence), Italy

[11]University of Bologna, Bologna, Italy

[12]INFN - Sezione di Bologna, Bologna, Italy

[13]ASI Space Science Data Center (SSDC), Rome, Italy

[14]INFN - LNF, Frascati (Rome), Italy

Abstract

The China Seismo-Electromagnetic Satellite (CSES) space mission—also known as *Limadou* in Italian—is a scientific collaboration between China and Italy that aims to investigate the structure and dynamics of the iono/magnetosphere, and in particular to study the possible correlation of perturbations to the occurrence of high-magnitude seismic events. The Chinese satellite houses the Italian *high-energy particle detector* (HEPD), an apparatus composed of a silicon tracker and a calorimeter system, which detects electrons and protons in the energy ranges 3–100 and 30–200 MeV, respectively. The satellite was launched on February 2, 2018; after a commissioning phase dedicated to the setting of an optimal configuration, HEPD is fully operational since July 28, 2018. After a general overview of the CSES-Limadou mission, this article presents the structure of the HEPD apparatus and describes the design and implementation of the software for the embedded computer system that is responsible for HEPD management. The onboard software runs on the digital signal processors of two electronic boards (CPU and data acquisition), monitors system status, handles instrument data acquisition, performs periodic calibration of the subdetectors, performs

[15]Department of Physics, University of Naples Federico II, Naples, Italy
[16]Italian Space Agency, Rome, Italy

Correspondence
Alessandro Sotgiu, INFN, Sezione di Roma "Tor Vergata", Rome, Italy.
Email: alessandro.sotgiu@roma2.infn.it

data compression, and communicates with the satellite. An overview of HEPD in-flight operations and performance is given in the last section.

KEYWORDS
data acquisition, digital signal processors, earthquake, embedded software

1 | INTRODUCTION

The China Seismo-Electromagnetic Satellite (CSES) space mission[1] is a scientific program dedicated to the monitoring of perturbations of electromagnetic fields, plasma and particles in the top-side ionosphere, induced by either natural sources and anthropogenic emitters. Its aim is also to study possible correlations between such perturbations and the occurrence of seismic events.[2] In general, the CSES mission investigates the structure and dynamics of the ionosphere, the coupling mechanisms with the plasma layers, the temporal variations of the geomagnetic field in both quiet and disturbed conditions, and the fluctuations in the charged particle fluxes. For these purposes, the Chinese satellite houses several payloads[3-6] for the measurement of the electric- and magnetic-field components, plasma density and temperature, and energetic particle fluxes and spectra.

CSES mission, also known as CSES-*Limadou* in honor of Matteo Ricci (a Jesuit and explorer of the XVI century, Limadou was his Chinese name), involves an international collaboration between China and Italy. The Italian team has been responsible for the design, construction, testing and managing of the high-energy particle detector (HEPD),[7] one of the two payloads dedicated to the measurement of charged-particle fluxes. The detector aims to observe particle precipitation from the inner Van Allen Belt, possibly induced by coupling to lito-ionospheric perturbations as a result of seismic activity.[8-10] Thanks to its wide energy range (3–100 MeV for electrons, and 30–200 MeV for protons), combined with the nearly polar orbit, HEPD can also provide a valuable contribution to the study of space weather phenomena and cosmic ray propagation.[11-14]

CSES-01 has been in orbit since February 2, 2018; after a commissioning phase aimed to set a working configuration,[15] HEPD has been fully operational since July 28, 2018. It is expected to be working for at least 5 years.

A second satellite (CSES-02) is currently in an advanced implementation stage, and is intended to provide wider coverage of the near-Earth electromagnetic and particle environment thanks to its orbit plane, the same as CSES-01, but with a 180° phase difference.

After a brief description of the HEPD apparatus and its electronic subsystem in Section 2, control and acquisition software packages are illustrated in Sections 3 and 4, respectively. An overview of HEPD's in-flight operations is given in the Conclusions.

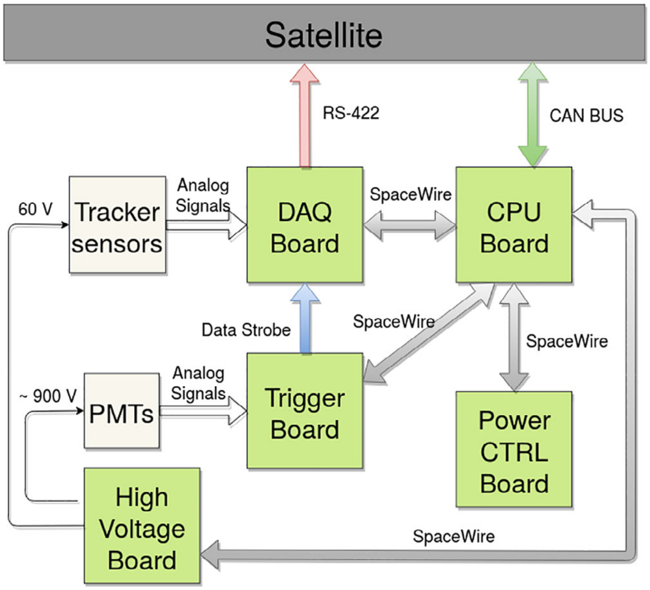
2 | THE HIGH-ENERGY PARTICLE DETECTOR

The HEPD detector is composed of a set of subdetectors designed to identify and measure electrons, protons, and light nuclei that cross the instrument, with the possibility to discriminate the trapped component of cosmic rays from those with an extra-terrestrial origin, through the reconstruction of the direction and energy of incident particles.

HEPD's sensitive subdetectors include:

- a **tracker system**, consisting of two double-sided parallel planes of silicon microstrip sensors, which is used to reconstruct the incident-particle trajectory;
- a **trigger system**, which is composed of a segmented plane of plastic scintillator bars—each one read out by two photo-multiplier tubes (PMTs)—and used for the generation of the trigger signal at the passage of a particle;
- a **calorimeter** for the measurement of the incident-particle energy, which is composed of two sections: the upper part includes 16 planes of plastic scintillators (each one read out by two PMTs); the lower part includes a plane of Lutetium-Yttrium Oxyorthosilicate inorganic scintillator crystals, arranged into a 3 × 3 matrix (each crystal is read out by a single PMT);

FIGURE 1 A general scheme of HEPD's electronic subsystem and power supply, with communication links between the electronic boards and satellite [Color figure can be viewed at wileyonlinelibrary.com]



- an **anticoincidence system**, including five planes of plastic scintillators that surround the calorimeter (each one read out by two PMTs), used to reject particles that enter the apparatus from outside the acceptance window, as well as those not fully contained within the calorimeter.

The outputs of the subdetectors describe an *event* (i.e., the detection of an incident particle) that must be digitized and processed. The result of this processing is sent to the satellite in order to be forwarded to the ground.

The electronics (HEPD electronics subsystem—ELS) that control the whole detector are described in Reference 16. Those of interest for this software description include: (1) the **central processing unit (CPU) board**, which controls the whole system and enables communication with the satellite, (2) the **data acquisition (DAQ) board**, which collects event information from the detector and processes it, and 3) the **trigger board**, which implements the trigger system and the readout of PMT data.

A schematic view of the HEPD-ELS and its data channels is shown in Figure 1, while a detailed description of the whole HEPD apparatus, including subdetectors and electronic boards, can be found in Reference 17.

2.1 The microprocessor

The computing power of the CPU board and DAQ board is provided by a digital signal processor (DSP). The analog device ADSP-2189M[18] has been chosen for both boards because of its extended operating temperature range, low power consumption, and high reliability in space environments, as tested in previous satellite missions such as PAMELA.[19-21]

This chip combines the ADSP-2100 family architecture with a set of peripherals; in particular, it includes two serial ports, a set of programmable input/output (I/O) lines that can be used as either hardware interrupts or control bits (flags), 192 kB of on-chip memory, an internal 16-bit data memory access port, and a byte DMA[1] (BDMA) port that can be used for external memories. The BDMA is able to transfer data while the DSP is operating normally, with a small impact on the DSP performance as it operates in steal mode (i.e., BDMA waits for the DSP to grant the bus, sending only one word of data, and then immediately releasing the bus for the DSP to continue). The ADSP-2189M implements a Harvard architecture[2] with a program memory composed of 32 kwords (24 bit) and a 48 kwords data memory (16-bit). Both memories are divided into pages; some are static and always available, while others are managed as overlays (pages of memory that share the same addresses and are identified by a page number), which must be selected before use, resulting in a small but cumulatively

[1] Direct memory access is a method that allows certain hardware subsystems to fast access the RAM without moving data through the CPU and possibly without interfering with its activity.
[2] The Harvard architecture is a computer architecture with separate storage and signal pathways for instructions and data. It contrasts with the von Neumann architecture, where program instructions and data share the same memory and pathways.

appreciable delay. In order to maximize performance, the always-available pages of both on-chip memories have been used to store the most frequently used functions and detector's calibration data. Overlay pages are used to store specific algorithms and to temporarily save processed data before delivery to the satellite. Programmable Wait-State generation has been used to cope with slow peripheral devices.

2.2 | Data communication links

HEPD's communication requirements imply the use of different types of data channels and protocols:

- The **data strobe** connects the trigger board to DAQ board. It is used to transfer digitized PMT data from the Trigger (responsible for readout) to the DAQ (responsible for processing).
- The **RS-422 link** connects the DAQ board to the satellite in order to transfer scientific data related to acquired events. This serial link has been adopted due to its simplicity and suitability for space payloads[22]). For redundancy purposes, two physical links (`Nominal` and `Redundant`) have been implemented.
- The **CAN bus**[23] is a multimaster serial bus used by all CSES payloads to communicate with the satellite. CAN messages have a priority: high-priority messages are transmitted with no delay, while low-priority ones are temporarily hold in case the bus is busy and automatically retransmitted after the end of the higher priority message. In the design of the CAN network, the satellite is the only master, while all other payloads on board CSES, including HEPD, are slave nodes. This means that HEPD itself is allowed to send messages on the CAN bus only as a response to a previous message from the satellite. The first bits of each CAN frame contain the identifier of the sender, as well as message priority bit, recipient node address, and message type. For redundancy purposes, two physical buses (CAN bus A and CAN bus B) are present, but only one bus is active at any time.
- The **SpaceWire Light**[24] connections are used by the HEPD CPU board to enable communication with other HEPD boards for housekeeping and configuration purposes. This fast local communication network is especially designed for spacecraft, and it has been selected for its high reliability and positive performance.[25] *SpaceWire Light* controllers are implemented in the field programmable gate arrays (FPGA) firmware of the electronic boards: each board exposes a number of 32-bit registers, such that the CPU board processor can read their status and configure them by means of commands. FPGA are well suited for space applications.[26]

2.3 | The CPU board

The CPU board is the subsystem that controls all detector operations and adds the timestamps to the data collected during acquisition and calibration activities (time information related to acquisition and calibration runs). It continuously collects the status of all other electronic boards via a *SpaceWire Light* link, and it also manages the communication with the satellite computer via CAN bus interface.

The CPU board has a symmetrical design with two identical and completely independent sections, which are provided for redundancy purposes. The main components of each part of the board are listed below:

- a DSP executes the program to manage all detector's in-flight operations, with routines for system diagnostics and configuration;
- an FPGA controller boots the board and provides interboard communication via the 32-bit registers mentioned above;
- two CAN bus transceivers and controllers connect the board to the satellite;
- a read-write FRAM[3] contains the DSP boot program of the board, as well as configuration parameters and status of the system before the last turn-off, and diagnostic and configuration applications;
- a read-only EEPROM[4] contains a recovery version of the DSP boot and application software to be used in case of failure of a normal boot from FRAM.

[3] Ferroelectric RAMs (FRAM, see Reference 27) are nonvolatile read/write memories similar to flash memories, a little bigger but with lower power consumption, faster writing operations, high robustness to *single event upsets*, much higher endurance to the write cycles than other nonvolatile technologies such as flash memories and EEPROMs.

[4] Electrically erasable programmable ROM.

2.4 | The data acquisition board

The DAQ board interfaces with the front-end electronics of the tracker system for processing and digitization of the analog signals. In addition, it performs detector calibration (both tracker and PMTs) upon request of the CPU board and provides compression and formatting of the acquired scientific data to be transmitted to the satellite. The DAQ board shows a quasi-symmetrical design: for redundancy purposes, it is composed of two identical parts that share a common interface with the tracker detector since the large number of analog data channels (4608) is impractical to duplicate.

The DAQ board architecture, shown in Figure 2, consists of the following components:

- a DSP running the DAQ software;
- a first FPGA (named *main FPGA*) used for safe boot of the DSP and tracker readout;
- a second FPGA (named *download FPGA*), which is used for transferring scientific data to the satellite via RS-422 link;
- two nonvolatile FRAMs to store two copies of the DSP software;
- a dual-port random-access memory (DPRAM), used as transit buffer between the main FPGA and DSP;
- a static random-access memory (SRAM), used as a FIFO buffer for final delivery of HEPD scientific data to the satellite;
- eight 12-bit analog-to-digital converters (ADCs) for digitizing the tracker sensor data.

The two nonvolatile FRAMs not only store the DSP code, but also allow software update after satellite launch. This update procedure is performed by the CPU.

When the board boots, it takes advantage of a feature of the ADSP-2189M, which automatically loads the software code from the first FRAM into the DSP program memory by using its *byte memory interface*. In case of failure, the second FRAM is automatically used. Some of the not-always available pages of the DSP are mapped to the DPRAM and used to exchange data between the main FPGA and DSP. The *byte memory interface* is also used to map the SRAM used as a buffer for sending data to the satellite.

2.5 | The trigger board

The trigger board is the system that collects data from the PMTs and selects particle events. Whenever an event is recognized, the trigger board sends a pulse to the DAQ board in order to collect tracker data of the event. Analog data from the PMTs are digitized by dedicated ADCs and then transmitted to the main FPGA of the DAQ board via data strobe link. For calibration and debugging purposes, the trigger board can also generate artificial trigger pulses. The board also measures the nonactive (dead) time and active (live) time of the detector, which are needed to calculate the real event rate starting from the measured one.

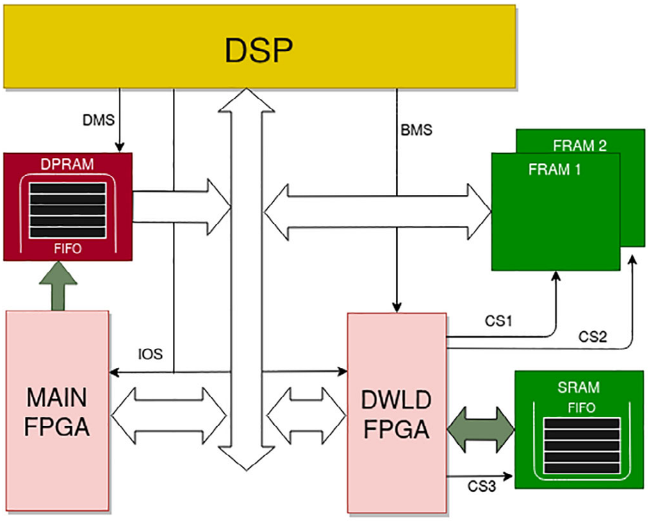


FIGURE 2 A block scheme of the DAQ main components showing the ports used by the DSP to access the external memories. DMS, IOS, and BMS are the chip selects of the data memory port, the I/O space, and the byte memory port, respectively. The access to the SRAM and the two FRAMs is handled by the download FPGA which enables the correct chip select (CS1, CS2, or CS3) depending on the address that the DSP needs to access [Color figure can be viewed at wileyonlinelibrary.com]

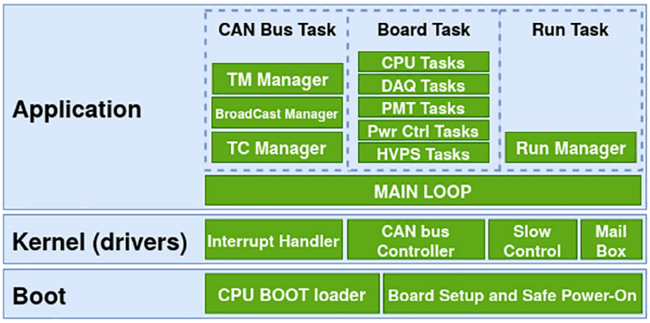


FIGURE 3 A scheme of the CPU software architecture [Color figure can be viewed at wileyonlinelibrary.com]

2.6 | Data flow of an event acquisition

When the trigger board and DAQ board have collected all the information related to an event, the former transfers such data to the main FPGA of the DAQ board through the data strobe link. At this point, all data are stored in the main FPGA. Data are then copied to the DPRAM, such that the DAQ-DSP can retrieve them all, compress tracker data (PMT data are not compressed due to their small size) and packet them. These packets are then transferred to the SRAM, which, in turn, is asynchronously read by the download FPGA. As soon as 4112 bytes are received, the download FPGA creates a frame to contain them by adding a header section and a trailing checksum, with final delivery to the satellite via RS-422 link. HEPD data are transferred to the ground by the satellite using an X band connection at a rate of 120 Mbps over the time intervals corresponding to the passage over five ground stations (all located in China).

3 | CPU BOARD SOFTWARE

The software run by the DSP of the CPU board manages the whole HEPD apparatus and communication with the satellite computer, also monitoring HEPD status. In particular, the in-flight operations of the CPU application are related to the management of messages from/to the satellite: broadcast information, telecommands, telemetries. In addition, the software configures, starts, stops, and monitors the calibration and acquisition sessions ("run"), as well as the status of each subsystem.

The architecture of the software is composed of different modules, grouped on different layers (see Figure 3):

- the *Boot layer* hosts the boot loader responsible for the correct initialization of the CPU, and the routines to power on and setup any subsystem;
- the *Kernel layer* hosts the drivers of the CAN buses, the slow control link and Mailbox message passing systems, and the interrupt handlers;
- the *Application layer* contains the main application responsible for HEPD working operations.

These modules are described in the following.

3.1 | The boot layer

The CPU board uses two different nonvolatile memory chips to store the application program: a writable FRAM and a read-only EEPROM. At power on, the *CPU boot* module instructs the DSP to load the application program and related data (such as previous boot status and configuration) from the FRAM memory[5] to its internal RAM. This is performed by means of the ADSP-2189M *byte data memory interface*, which uses the BDMA port. An FPGA watchdog timer is set to force the boot from the EEPROM in case of boot failure; if the upload from the FRAM is successful, the watchdog timer is disabled.

[5] For redundancy, both the application program and related data are duplicated and saved on two different locations of the FRAM.

The two modules of the boot layers are the *CPU boot loader* and *board setup and safe power on*. After the CPU-board boot, the newly loaded *board setup and safe power on* module runs to complete the power on procedure for HEPD subsystems.

The power on operation must follow a proper sequence; the first step is to power on and check all the electronic boards, one at a time: each board is enabled and its status and basic functionalities are immediately checked by sending commands via the slow control link (see Section 3.2.1); if the board booted correctly, the CPU powers on the next board following a defined order: power control board first, then trigger board, DAQ board, and high voltage board.

If the power on of all boards completes successfully, HEPD is set to either NOMINAL or SAFE mode, depending on the value of a specific parameter (*boot_mode*) saved in the configuration data.

When the system is configured to **NOMINAL mode**, a series of commands are sent to the *high voltage power suppliers* (HVPS) to gradually raise the high voltages needed by the sensors to operate properly (up to 900 V for the PMTs connected to the calorimeter and about 60 V for the tracker sensors), while checking the high voltage board error status. This is the only mode in which the system can perform acquisition of real events and calibration of the sensors.

SAFE mode is the state of the system when all the electronic boards are powered on, but the voltage applied to the sensors is 0 V. In this state, detectors can be readout to check the whole acquisition chain, up to satellite transmission. This state is reached either on purpose for testing (by changing the *boot_mode* parameter in the configuration data), or automatically when the last HEPD power off has been performed incorrectly. In the latter case, the normal behavior (NOMINAL mode) is automatically restored after a reboot of HEPD by the application of the correct power off and reboot sequences.

Once HEPD is set in either NOMINAL or SAFE mode (called OPERATIONAL modes), the software enters the *main loop* function, which is located on the *application layer* (see Section 3.3). The HEPD state is periodically changed by the satellite (by means of telecommands) to **STAND-BY mode** in order to temporarily suspend the acquisition. During this time, satellite attitude adjustments are performed and all payloads are requested to be inactive, so they are switched to a minimum functionality to reduce power consumption. This primarily happens when traveling over the polar regions (±65° of latitude), where seismic phenomena are of limited interest and magnetospheric phenomena are more easily perturbed by the higher connection to the interplanetary space. The main processes and states of the HEPD system are shown in Figure 4, together with the main telecommands automatically sent by the satellite to change the state. *Wired commands*[6] and CAN bus telecommands sent by the satellite are marked by red and blue arrows, respectively.

3.2 | The kernel layer

The Kernel layer hosts the software used by the other modules to transfer data, and implements the various communication protocols. As the CPU is responsible for the house-keeping activities and configuration of the various subsystems, different communication protocols have been implemented:

- the *Slow control link* protocol allows all the HEPD electronic boards to exchange data via the *SpaceWire Light* connections,
- the *CAN bus link* protocol is used for communication between the satellite and HEPD,
- the *Mailbox* protocol allows the DSP of the DAQ board to communicate with the DSP of the CPU board.

3.2.1 | Slow control link

The slow control link, used for communication between different HEPD electronic boards, is implemented with *SpaceWire Light* technology, and the controllers are implemented in the FPGA firmware of each electronic board.

The slow control protocol relies on two types of commands: `remote_read` and `remote_write`. Each command message is composed of a 10-byte structure containing the type of operation requested, the address of the remote resource, and the data section.

[6] Physical wires used to power on/off the system and to set/reset the standby mode.

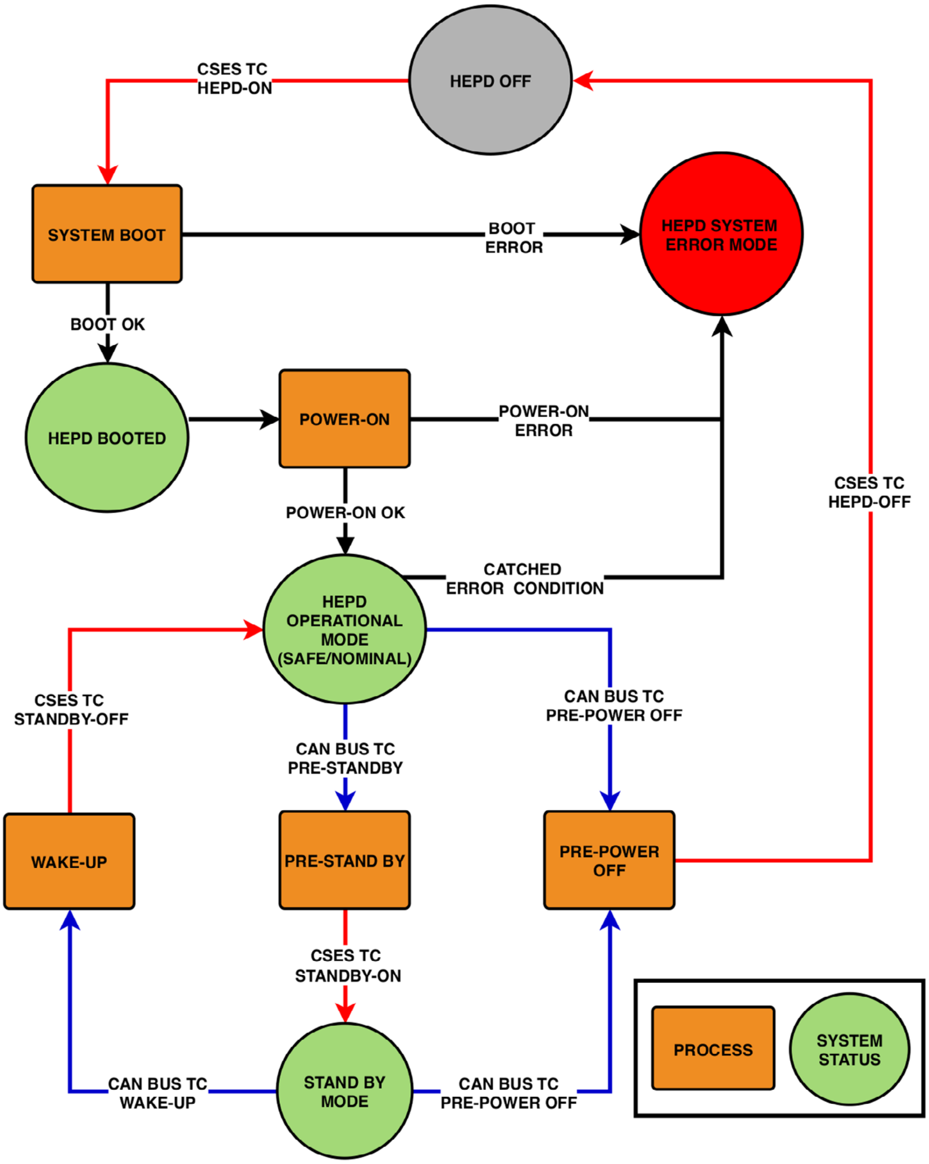


FIGURE 4 Main processes and states of the HEPD system, and main commands for state transitions. Wired commands and CAN bus telecommands are marked by red and blue arrows, respectively. Black arrows represent an automatic change of state, induced by the success (or failure) of the current system operation [Color figure can be viewed at wileyonlinelibrary.com]

The CPU board can configure and monitor the status of the other boards through the local FPGA, which accesses the FPGAs of the other boards via *SpaceWire Light* links. Each board FPGA exposes a number of 32-bit registers that are identified by specific I/O addresses on the local FPGA.

In detail, the slow control drivers have been implemented to provide the following API functions:

- `spw_read()` reads a 32-bit *SpaceWire Light* register;
- `spw_write()` writes a 32-bit *SpaceWire Light* register and checks the written value;
- `spw_post_write()` writes a 32-bit *SpaceWire Light* register (no check of the written value);
- `swp_testChannel()` tests a specific channel for correct behavior.

3.2.2 | CAN bus controller

The satellite sends three different types of messages through the CAN bus:

- Telecommands messages;
- Telemetry sequence messages (slow poll/fast poll or telemetry data);
- Broadcast messages.

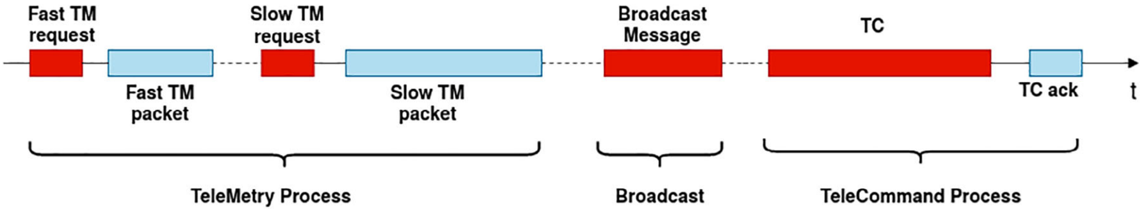


FIGURE 5 Scheme of a transmission on the CAN bus link. In red the frames sent by the satellite, in cyan the response from HEPD [Color figure can be viewed at wileyonlinelibrary.com]

Figure 5 shows how the transmission between the satellite and CPU board of HEPD takes place during in-flight operations. Red blocks represent messages sent from the satellite, while cyan blocks represent the response of the DSP.

Telecommands (TC) are messages that control the correct power on sequence, configuration and management of HEPD. TCs can be divided on the base of the number of CAN frames that compose the command: there are *single-frame TCs* and *multiframe TCs*. Single-frame TCs have a size of 8 bytes, while multiframe messages can amount to 9, 57, 121, or 249 bytes. After receiving a TC message, the CPU first checks its integrity and acknowledges reception by sending back a 6-byte message, then it executes the command (see Section 3.3.1).

The satellite periodically sends a **telemetry** (TM) request to the DSP in order to collect the status of HEPD subsystems. The DSP interrogates the subsystems through the slow control channel to obtain their status, then it sends this information back to the satellite by using TM messages. There are two types of TM messages:

- *Fast Poll* (20 bytes long) are sent at a frequency of 1 message per second,
- *Slow Poll* (100 bytes long) are sent with a lower frequency (every 8 s), but contain more information.

TM data are sent as a sequence of CAN frames having a fixed dimension (10 bytes) and structure, and contain different information depending on the type (see Section 3.3.1). The satellite downloads all the acquired TM to the ground station once a day. This rate is adequate to respond to malfunctions or unexpected behavior. Part of the TM data is also sent to the DAQ board at both the start and stop of each acquisition session, so as to be incorporated in the scientific output data.

All payloads are sent by the satellite four types of **Broadcast messages** containing satellite information such as position, velocity, time and attitude. Depending on the type, Broadcasts messages can be either single-frame or multiframe. These messages are sent every second, except for the single-frame broadcasts carrying satellite time, which are sent every 8 s.

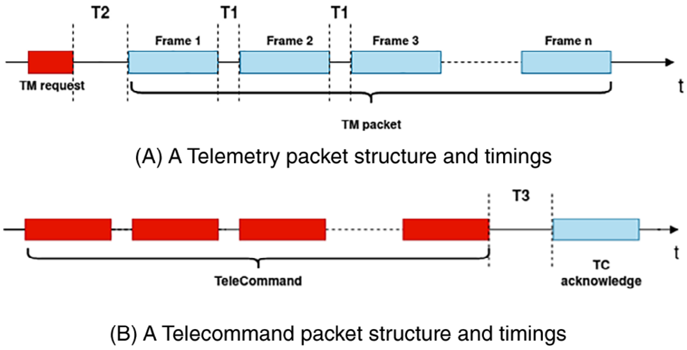
Broadcast information is used to identify the current orbital zone in order to plan the next operation (see Section 3.3.3), and also sent to the DAQ board to provide essential information for physics analysis of scientific data.

3.2.3 | CAN bus controller implementation

One of the tasks of the CPU-board DSP is actually to implement the CAN bus protocol on the CAN bus link. A new driver has been developed in order to fulfill the time constraints imposed by the satellite specifications. The driver performs the following tasks:

- initialization and configuration of the CAN controllers (setting of the bit-rate, and of the incoming message allowed);
- management of both CAN bus channels;
- implementation of the HEPD CAN bus protocol carrying the aforementioned telemetries, telecommands, and broadcast messages;
- support for a debug protocol that enables read/write accesses on the slow control channels.

To access and configure the two CAN bus controllers, the FPGA separately exposes the registers of the two controllers, such that the DSP can access them by using its I/O space interface. During this phase, the DSP activates a network filter



(A) A Telemetry packet structure and timings

(B) A Telecommand packet structure and timings

FIGURE 6 Scheme of the telemetry (A) and telecommands (B) transmission between the satellite and HEPD detector. T1 is the delay between two consecutive frames of a TM packet and must be $0.2\text{ms} < T1 < 0.6\text{ms}$. T2 and T3 are the delays between the TM request and the first frame of the HEPD response, and the delay between the last frame of a TC sequence and the HEPD response, and must be $0.2\text{ms} < T2/T3 < 2\text{ms}$ [Color figure can be viewed at wileyonlinelibrary.com]

on the FPGA in order to accept only messages directed to HEPD, and it sets up the transmission protocol parameters required to communicate with the hosting satellite. Both CAN bus controllers must be configured, despite only one bus is active (i.e., with bit exchange) at any time. The communication protocol with the satellite provides for a switch to the redundant bus in case of incorrect communication. In particular, if the satellite sends a TC but does not receive the acknowledgment in time (or if the acknowledgment message is erroneous), it tries to send the TC once again. If the error persists, the satellite switches to the redundant bus, trying to send the command for the last time before reporting the error to the ground station.

FPGA firmware implements the CAN controller to know when a message has been sent from the satellite to the DSP, and on which CAN bus. In order to make the DSP read the message from the bus, the FPGA asserts a physical line connected to the DSP, thus triggering a hardware interrupt. Each CAN bus has its own physical line, while the interrupt handler is shared.

Figure 6(A,B) show the time windows allowed for the response to TM data (composed of multiple CAN frames) and to TC (single or multiframe), respectively. According to Figure 6(A), T1 is the delay between two consecutive frames of a TM packet, and T2 is the delay between the TM request from the satellite and the first frame of the HEPD response; while T3 (Figure 6(B)) is the delay between the last frame of a TC sequence sent from the satellite and the HEPD response. The range of values allowed for these three delays have been defined by the satellite manufacturers because the same link is shared by all the payloads to communicate with the satellite, and are $0.2\text{ms} < T2$, $T3 < 2\text{ms}$ and $0.2\text{ms} < T1 < 0.6\text{ms}$. Compliance with these times is guaranteed by the interrupt handler driver which manages and uses the DSP internal timer to schedule with high precision the start of the transmission of CAN bus frames.

The interrupt handler uses the `sum_calc` function to check the integrity of the message content and the `start_timer` function to verify timing compliance. Then it uses the `send_ack` function to send back an acknowledgment. After these preliminary operations, the handler identifies the type of the received message and acts accordingly.

In case of a telecommand, the DSP timer is set, and the command is stored in a circular buffer. The commands will be executed in a FIFO order when the `CAN bus_task` (see Figure 7) is invoked in the *main loop* application.

A telemetry request (slow/fast poll) is handled in a way similar to a telecommand: timers are set to ensure the T1 and T2 timings to be within the correct range of values (see Figure 6(A)). The telemetry messages are saved in the DSP internal RAM, and they are updated in the *main loop application*every time a board task is requested. When the timer goes off, all the telemetry messages are sent.

In case of a Broadcast message, no response is needed and the handler routine just saves the data to the internal RAM for the *main loop application* to use it (e.g., determining satellite position).

3.2.4 | Mailbox protocol

The DSP on the CPU board (CPU-DSP) and the DSP on the DAQ board (DAQ-DSP) exchange data by using a *Mailbox* protocol (MBOX). The main FPGA of the DAQ board provides two 32-bit data registers and two control bits. The CPU-DSP accesses these registers via the slow control link, while the DAQ-DSP uses an I/O address space.

The two registers are:

- `DAQ2CPUMBOX`: used by the DAQ-DSP to post data for the CPU-DSP;
- `CPU2DAQMBOX`: used by the CPU-DSP to post data for the DAQ-DSP.

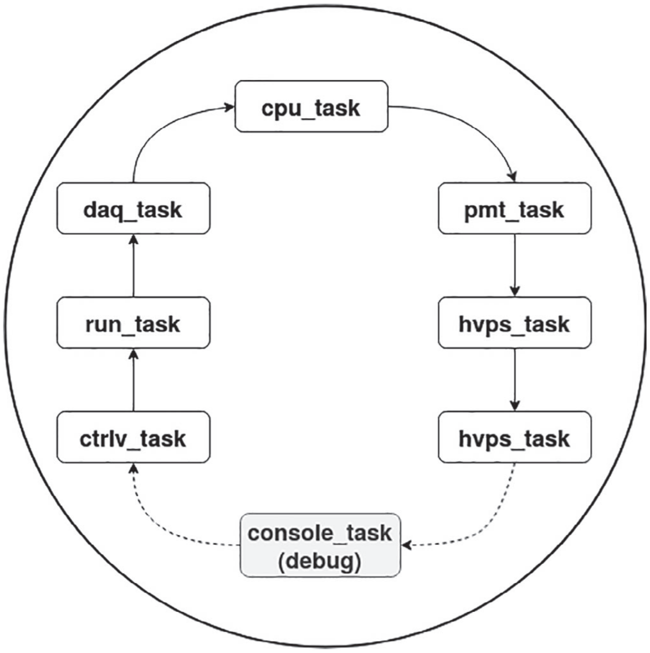


FIGURE 7 Block scheme of the main loop running on the DSP of the CPU board

The content of the two data registers is structured as follows: the 16 most significant bits are the command ID that specifies the operation, while the lower 16 bits are an optional parameter of the command. In order to access the data register, its corresponding control bit is tested, set when data have been written to the register, and reset when data have been read from. This prevents overwriting of data not yet read and concurrent access to the register.

The CPU-DSP is the master of the MBOX protocol, thus the DAQ-DSP can only write responses on the Mailbox after a previous CPU-DSP command. This response message contains the received message. The CPU-DSP checks the returned value and, if correct, continues the sequence.

The MBOX protocol messages that the CPU board sends to the DAQ board can be either software upgrade commands or application commands. **Software upgrade commands** allow for the upgrade of the DAQ software. Indeed, an update of the CPU-DSP can be achieved by sending proper CAN bus TC messages, while the DAQ-DSP cannot access the data content of the TC messages by itself, thus the use of the mailbox is needed. The DSP software is stored in the FRAMs, the aforementioned set of commands allows to rewrite part or all of the FRAMs contents.

The *software upgrade* protocol is a relatively slow procedure, since the size of the DAQ-DSP program is about 120 kB and the mailbox protocol sends messages that carry a payload of 2 bytes of data each. Moreover, each message must be requested by a CPU-DSP command. The update of the whole DAQ-DSP software requires about 2 min; for this reason, this procedure is considered very risky and to be applied only in case of critical issues. For security reasons, a minimal version of the DSP program (called the *miniboot*) is also stored in a nonvolatile memory of the main FPGA of the DAQ: this memory is not writable by this procedure. In this way, in case an error corrupts the FRAM content during the upgrade operation, it is always possible for the DSP to boot from the FPGA and repeat the software upgrade procedure.

The **application command** group is related to the activities the CPU-DSP requests in order to perform a self-test of the DAQ board memories, to start the acquisition/calibration operations, and to perform preliminary DAQ power off operations, which are required to properly close the current acquisition phase before hardware power off. The execution of each mailbox command depends on the current DSP state: each one has a list of allowed commands.

3.3 | The application layer

The main loop application in the application layer is where the acquisition of scientific data from the sensors actually takes place. A well-defined sequence of tasks are initialized and scheduled by two main functions: `hepd_schedulerInit` and `hepd_schedulerLoop`; the former function initializes the sequence of tasks that will be invoked in the main loop, while in the latter function the sequence of the initialized tasks is invoked in an infinite loop.

The actions performed by the main loop have essentially the following purposes:

- continuous check of the status of the whole system;
- start and stop of the acquisition run;
- periodic start of the calibration phases;
- continuous management of satellite communications (via the CAN bus).

and the loop is executed until an unrecoverable error occurs.

The *main loop* calls a number of helper functions grouped by their functionality as *CAN bus tasks*, *board tasks*, and *run tasks*.

3.3.1 | CAN bus tasks

By calling the functions implemented in the CAN bus driver, these tasks are responsible for the management of the *broadcast packets*, the *fast poll* and *slow poll* control sequences (to periodically send telemetry information to the satellite upon its request), and for the execution of the received telecommands (by calling the corresponding function in the TC manager module).

Several TCs have been developed in order to completely configure and manage HEPD. They can be classified into three groups based on their functions, each group with a specific group identifier code:

- the **system control** single-frame TC messages can be sent when the system is not acquiring data or performing calibration (IDLE state, see Section 3.3.3), and are applied immediately; these TCs are used to power on/off sections of the detector, to start/stop DAQ and to save/load FRAM configuration;
- the single or multiframe **configuration** TC messages can be sent at any time, and are used to configure the HEPD, in particular HVPS voltage and sensor parameters. Configuration TCs are received and kept on hold; a system control TC message is subsequently required to apply the changes;
- the single or multiframe **test** TC messages can be sent when the system is not acquiring data or performing calibration (IDLE state), and are applied immediately; most of them have been developed to test the CAN bus, the RS-422 bus, and the slow control communications.

After receiving a TC message, checking its integrity and sending the acknowledge reception to the satellite (see Section 3.2.2), the CAN Bus task is responsible for the TC identification and execution accordingly to its command group.

The satellite periodically sends a *TM request* to CPU board in order to collect the status of HEPD subsystems. The CAN Bus tasks are responsible for the transmission of the information back to the satellite by using fast and slow telemetry messages (see Section 3.2.2). Depending on the type, the information sent includes:

- the status and error registers of each electronic board (both fast and slow poll);
- the readout of the temperature sensors placed on the CPU board and trigger board;
- current run information (run number, run time, and orbital zone);
- monitored values of high voltages of the sensors;
- the last received *telecommand* (in order to reconstruct the sequence of operations performed by HEPD);
- CPU information (boot source, CPU status before/after reboot, CPU time).

Finally, the CAN BUS tasks are responsible for the management and storage of the different broadcast messages. Position information is needed in order to either plan the next operation to be executed (acquisition sequence or calibration) or to properly configure the detector, for example, because of the different rate of incident particles along the orbit (see Section 3.3.3). For a complete analysis and reconstruction of the event, information such as time, position, and attitude of the satellite is essential. The complete broadcast data sent by the satellite are also forwarded to the DAQ board in order to include them into the scientific data packets.

3.3.2 | Board tasks

The *board task* group of helper functions provides for:

- monitoring the main loop and updating specific board telemetry data;
- enabling the whole board;
- powering on the board;
- checking the board;
- disabling the board for the power off procedure.

In detail, the main loop cyclically calls the following tasks to operate on each electronic board:

- the `cpu_tasks` executes all the internal *CPU board* checks and fills up TM data structures with CPU board information;
- the `pmt_tasks` takes care of the *trigger board* management (switch on/off, channel testing, configuration) and updates the TM data structures with trigger board information;
- the `hvps_tasks` takes care of the *high voltage power supply board* management (switch on/off, channel testing, configuration, system board checking, communication) and updates the TM data structures with HVPS board information;
- the `pwr_ctrl_tasks` takes care of the *power control board* management (channel testing, configuration, system board checking, power on/off of the other boards) and updates the TM data structures with power control board information;
- the `daq_tasks` takes care of the *DAQ board* management (switch on/off, communication channel testing, configuration) and updates the TM data structures with DAQ information.

During the execution of the board control tasks, as the CPU interfaces with each electronic board (thus ensuring complete control and monitoring of the whole detector), the CPU also checks the status of the corresponding subsystem by reading specific status registers mapped in the FPGA of each board through the slow control protocol data flow.

At the same time, the CPU updates the configurations and collects information from the other boards, which will be sent to the satellite via the CAN bus link by using telemetry messages (`canbus_task`).

3.3.3 | Run tasks

The `run_task` is the only helper function of the Run Task group: it deals with the starting, stopping, and configuration of the acquisition and calibration runs. The main operations executed during this task are summarized as follows:

- identification of the orbital zone from the satellite position (latitude and longitude) received from the last satellite Broadcast message (this identification process will be detailed in 4.2.1);
- configuration of the acquisition by sending the orbital configuration parameters to the trigger board and DAQ board via the slow control links;
- automatic starting and stopping of acquisition and calibration runs, depending on the orbital zone configurations and satellite positions, and on the predefined run duration;
- sending of auxiliary information, such as broadcast data or HEPD configuration to the DAQ in order to include them in the scientific output data.

The `run_task` activity is implemented as a finite state machine (FSM). The possible current states of the FSM are `RUN` (for acquisition runs), `CALIBRATION` (for calibration runs) and `IDLE` (when HEPD is not acquiring data due to either configuration activities or standby mode). When HEPD is working in manual operation mode (e.g., during ground operations or tests) the current and target states of the FSM are the same and reflect the received telecommand (START/STOP ACQUISITION, START/STOP CALIBRATION). When HEPD is working in automatic operation mode, the target state is set to AUTOMATIC and, according to orbital zone configurations and calibration frequency, acquisition runs (current run state = RUN) and calibration runs (current run state = CALIBRATION) are performed periodically. The (target and current) FSM execution state is stored in the CPU board *status error register* and can be checked in telemetry data (see Section 3.2.2).

4 | DAQ BOARD SOFTWARE

As already mentioned (see Section 2.4), the DAQ board is responsible for the acquisition of sensor data, data compression and formatting, and transmission of scientific output data to the satellite in order to download them to the ground. The operations of the DAQ board are based on the interaction between the DSP and the two FPGAs (main and download) installed on the board.

The main FPGA initializes the DAQ board and selects the boot memory for the DSP. Then it starts the DAQ FSM on the DSP that controls the acquisition operations.

4.1 | The finite state machine

The DSP FSM is composed of eight states and is shown in Figure 8.

The DSP saves its state in a specific FPGA register (`DSP_STATUS`) that the FPGA monitors. Every unexpected change of the register causes a restart of the DSP. In order to avoid the FSM to block indefinitely, a watchdog implemented in the FPGA monitors the time the DSP spends in one of the active states (i.e., not `IDLE`, `INIT`, or `READY`) and restarts the DSP in case of need. The average time spent in each state was measured during the functional tests of the apparatus. The watchdog timer was then set by multiplying this average time for a confidence factor.

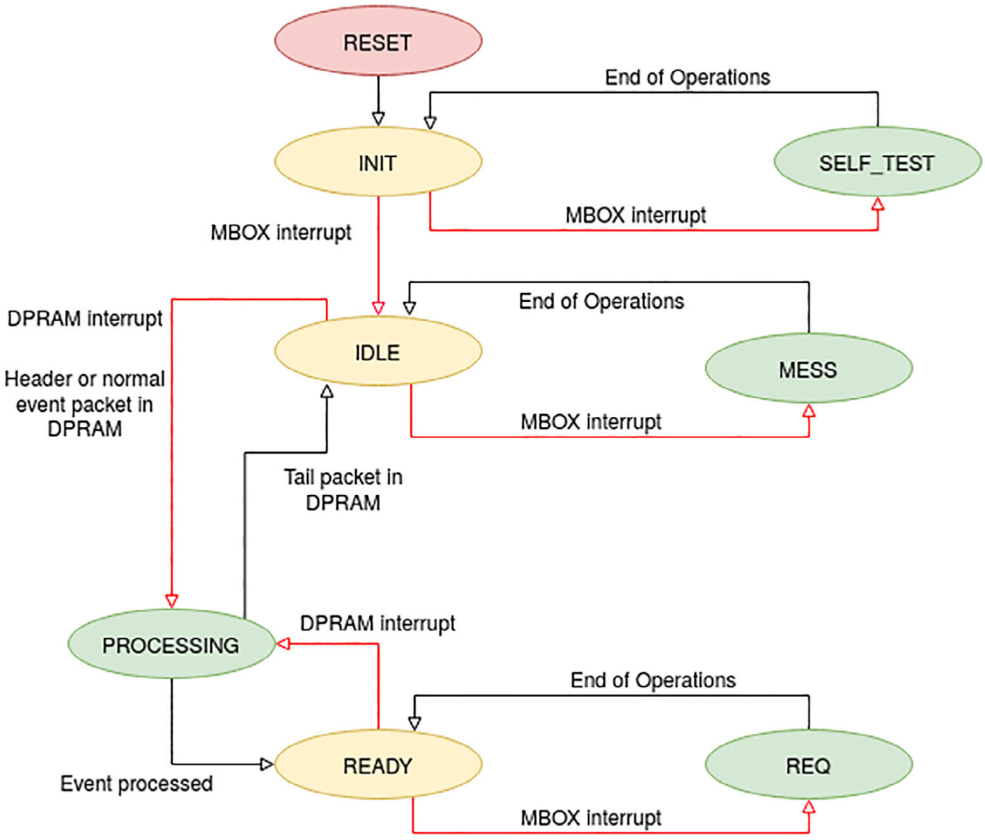


FIGURE 8 Finite state machine for the DSP of the DAQ board [Color figure can be viewed at wileyonlinelibrary.com]

The completion of the current operation causes a change of state of the FSM. Moreover, the main FPGA controls two external interrupt lines (red arrows in Figure 8) that trigger a change of the DSP state: interrupt line `MBOX_IRQ` is used when a new command from the CPU is received in the Mailbox registers and must be executed; while interrupt line `DPRAM_IRQ` notifies that a new event has been acquired and stored in the DPRAM in order to be processed by the DSP.

In the `INIT` state, the DSP has reduced functionalities and is sensitive to the `MBOX_IRQ` interrupt only, which informs that a new command has been sent by the CPU board. If the content of the CPU command is the request for an integrity test, the FSM changes to the `SELF_TEST` state, otherwise the FSM changes to the `IDLE` state. In the `IDLE` state, the DSP waits for a `DPRAM_IRQ` interrupt to start an acquisition session. Before sending an interrupt, the main FPGA always writes a command packet to the DPRAM so to instruct the DSP to perform a specific operation on it. The FSM state is then changed to `PROCESSING`, where the packet is analyzed and the requested operation is executed. The first command packet is actually a *header packet* containing all the detector configurations, previously set by the CPU during the execution of the `run_task` in the Main loop. These configurations have been sent to the DAQ board so as to be included in the scientific data temporarily saved in the SRAM, before being sent to the satellite. After this first packet is executed, the FSM state is changed to `READY` and the acquisition session is started. Different types of header packets can be used to start an acquisition run, a calibration run or a debug run.

During an acquisition run, a trigger signal is generated by the trigger board every time a particle crosses the instruments. The main FPGA starts the readout of the silicon detector channels, and the trigger board sends PMT data to the DAQ board via the data strobe link. Once this transmission is finished, the main FPGA writes all the scientific data of the event to the DPRAM and raises the `DPRAM_IRQ` signal, which causes the DSP to read and process data. Actually the DPRAM acts like a FIFO buffer for command/event packets, which allows the main FPGA to asynchronously write up to eight packets. This allows to reduce the dead time of the apparatus, and consequently to extend the maximum observable rate of particles. Once eight events are written in the DPRAM, the acquisition of new triggers is stopped by the DAQ until the DSP completes the processing of the first one.

During a calibration run, the trigger signal is not generated by a particle impinging on the detector, but by the trigger board itself at a constant frequency. In this case, the FPGA of the DAQ board performs the same operations as in a real acquisition, but the DPRAM packets contain the output produced by the sensors without signals generated by the passage of a particle, and they are therefore a measurement of the electronic noise.

A *tail* command packet is finally used to stop the current acquisition sessions: when it is executed the FSM state is changed back to the `IDLE` state.

4.2 | Acquisition software

The acquisition software includes all the procedures performed by the DAQ-DSP when it is in the `PROCESSING` state, that is, for a calibration or an acquisition run. Several routines have been developed for sensor calibration and event compression.

4.2.1 | Calibration runs

Calibration runs are special procedures performed at an adjustable frequency (currently once per orbit). Before starting a calibration run, the CPU-DSP checks the satellite position from the broadcast data and decides if it is possible to perform the calibration, in order to assure at least one calibration for every orbit. These runs are allowed only in the near-equatorial region[7] (excluding the SAA[8]), where the event trigger rate is lower and therefore the possibility of having real events during the calibration procedure decreases. The in-flight calibration procedure is also required in order to perform the data compression of the event as discussed in Section 4.2.2.

Each silicon channel (and also each PMT channel) fluctuates around a mean value (*pedestal*) with a certain variance (σ^2). The calculation of these two values is the aim of the online calibration. For this purpose, a fixed number of emulated events is acquired, as described below. The calibration algorithm can be described by the following four-step procedure:

[7] The near-equatorial region is defined as the region with a latitude value in the range $[-33°;+33°]$.

[8] The South Atlantic Anomaly (SAA) is an area where the Earth's Inner Van Allen radiation belt is closer to the planet surface, leading to an increase of the energetic particle fluxes. In the DSP software, it was approximated as a rectangular region defined by latitude in the range $[-50°;+0°]$ and a longitude in the range $[-90°;+20°]$.

1. **Pedestal calculation:**
   N false triggers are used to evaluate the pedestal of each channel. Assuming ADC_{ij} is the signal registered on the channel i for the event j, the pedestal of channel i is defined as $\mathrm{ped}_i = \frac{1}{N}\sum_{j=1}^{N}\mathrm{ADC}_{ij}$.
   Due to the limited DSP RAM, it is not possible to keep in memory the information of all the events at the same time. For each channel i, when the event l is acquired, only the partial sum $S_i = \sum_{j=1}^{l}\mathrm{ADC}_{ij}$ is saved in one of the overlays pages of memory. The pedestal is then calculated after the last event is acquired, by dividing the partial sum by N. The same consideration is true for the calculation of the following three steps of the calibration.
2. **Raw sigma calculation:**
   N false triggers are used to evaluate the channel fluctuation to characterize channel stability. Raw noise is calculated according to:

   $$\sigma_i^{\mathrm{RAW}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N_j}(\mathrm{ped}_i - \mathrm{ADC}_{ij})^2}, \tag{1}$$

   σ^{RAW} parameter is also used to identify *dead* channels (not working channels for which the σ^{RAW} value is compatible with that measured when the voltage of the silicon sensors is not enabled) and *hot* channels (noisy channels for which the σ^{RAW} value is greater than a configurable threshold), in order to exclude them in the following calibration steps.
3. **Corrected sigma calculation**:
   N false triggers are used to determine and subtract the noise that is common to all the silicon channels related to the same silicon sensor. Being the same for all channels, it can be estimated as follows:

   $$\mathrm{CN}_j = \frac{1}{N}\sum_{j=1}^{N_j}(\mathrm{ADC}_{ij} - \mathrm{ped}_i), \tag{2}$$

   where N_j is the number of good channels (excluding dead and hot channels). Once the common noise calculation is performed, it is possible to calculate the intrinsic noise of each channel without the contribution of the common mode noise as:

   $$\sigma_i^{\mathrm{TRUE}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N_j}(\mathrm{ADC}_{ij} - \mathrm{ped}_i - \mathrm{CN}_j)^2}. \tag{3}$$

4. **Bad working strip detection:**
   the last step of the calibration is used to identify those strips that do not have a Gaussian distribution of the signal in empty events. The estimator chosen to determine whether a channel is Gaussian or not, is the fraction of events giving counts outside $\pm$ 3 σ^{TRUE}. These strips, plus the dead ones, are the only strips that are not considered as a cluster seed in the compression algorithm. Noisy strips, with Gaussian behavior, are perfectly suitable to trigger a cluster.

Since the ADSP uses 16-bit words for data memory and the values of the σ^{TRUE} are usually quite small numbers (4–5 bit), in order to reduce the size of the calibration event the three most significant bits of the sigma variable are used as flags to report whether the current channel is dead, noisy or not having a Gaussian behavior.

As for the calibration of the PMTs, only the first two steps of the calibration procedure are applied, providing the pedestal and the raw sigma values of each PMT.

Once the calibration is completed, the pedestals and sigmas of each channel are saved on the two nonvolatile FRAMs for preserving them when the DAQ is turned off at the poles. These calibration data are loaded during the boot procedure, and also saved on the DSP internal RAM since the DSP frequently needs to access them to perform data compression, thus achieving a better computing performance. The calibration data are also sent to SRAM for future transmission to the satellite (see Section 4.2.3) as an event packet.

It is worth noting that all the parameters used in the calibration procedure (such as the number of false triggers, the thresholds used to define dead, noisy or not Gaussian channels, the number of calibrations per orbit or the region where the calibration is allowed) can be modified in any moment by means of a configuration TC.

4.2.2 | Acquisition run

HEPD spends most of CSES orbit time performing acquisition runs. In this case, the software operations concern the event data compression, formatting, and transmission.

The HEPD trigger rate amounts to an average value along the orbit of ≃100 events per second. This rate and a daily working time of about 17.6 h result in approximately 6×10^6 events/day. Since each event is stored in a packet of about 9.5 kB, HEPD produces a total of up to 57 GB of raw data per day. The satellite reserves for HEPD only 50 GB per day, thus raw data need to be compressed. As 97% of the raw data comes from silicon tracker planes, only these are subject to compression.

Each silicon plane is composed of two sensitive layers located on the upper and lower side of the silicon wafer, respectively. Each layer consists of implanted aluminum microstrips (strip pitch of 182 μm) that run throughout the entire width/length: the two layers have different strip orientation, such that X and Y coordinates of the incident particle can be identified. When a particle traverses the plane, just a few strips in each layer are involved (which constitute a *cluster*); the two strips that collect a higher signal are called the *seeds*. Actually the number of strips that compose a cluster is a configurable parameter that can be set to three, five, or seven by a specific TC: ground acquisition with cosmic muons and proton beam tests[28] showed that five strips are adequate to the purpose. When an event occurs, raw data from the silicon planes are read out and digitized; the compression algorithm then analyzes them in order to keep only significant data. The data compression routine is a custom implementation of the *zero-suppression* algorithm and starts with the evaluation and subtraction of the common noise, as described in (2). Then, in each layer, the strips with a value greater than their pedestals+3σ (with pedestal and sigma values of each strip provided by calibration and stored in the DSP memory) are searched for. When a channel with a value above the threshold is found, the program looks at the adjacent strips in order to identify the one with the highest value (the *seed* of the cluster). After that, the seed index and the content of the seed channel, together with the previous and the following two channels, are saved. In case of single event detection, four clusters (2 planes × 2 layers) are expected, but more clusters are possible due to statistical fluctuations of the electronic noise. The overall performance of the algorithms reduces silicon data to about 1.5% of the original raw format.

Unlike calibration runs, during which only the final results of the calibration (i.e., pedestals and corrected sigma values computed after the acquisition of 1024 simulated triggers) are transmitted to the satellite, in acquisition runs each event is transmitted to the SRAM as soon as it is compressed and formatted.

4.2.3 | Data formatting and transmission

Until a tail command packet is received, the DSP stores the results of data processing to its internal memory and transfers them to the SRAM by using the internal BDMA circuit, thus interleaving its transfers with normal system bus usage. As new blocks of data are produced by the DSP, they are queued to the BDMA in order to be sent as soon as possible to the SRAM. This sequence of blocks is managed by a FIFO list of block descriptors, each one containing the information about the starting address of a data block in the local RAM, its size and its destination address in the SRAM. When the transfer of a block completes, the BDMA notifies the DSP by sending it an interrupt. The interrupt handler routine `BDMAOp` on the DSP removes the first block descriptor from the list and retrieves from it the parameters needed to restart the BDMA transfer of the corresponding data block. When the data are in the SRAM, they must be sent to the satellite in order to be forwarded to ground.

Satellite specifications require that data sent through the RS-422 link are in frames of 4124 bytes. Data on the SRAM buffer are now just a stream of bytes, and the *download FPGA* divides them in chunks of 4112 bytes, each enclosed between a header and a trailer. The frame header contains the ID code of the HEPD payload and a progressive index, while the trailer contains a checksum. Packet building is spooled to this independent FPGA in order to allow the DSP to continue the acquisition of other events, thus reducing dead time.

Before a shutdown of the DAQ board, the contents of the SRAM (which is a volatile memory) must be transferred in order not to lose data that have not yet been transferred to the satellite. For this reason, the CPU sends a specific mailbox command to the DAQ-DSP to notify the imminent power off of the system. The current run is then stopped, and the last processed event is copied to the SRAM. The last frame is completed with zeroes in order to reach the correct size, and then all data in the SRAM are transferred to the satellite. This operation is repeated at every passage to the poles, where the HEPD detector is put in standby mode and the DAQ board is powered-off.

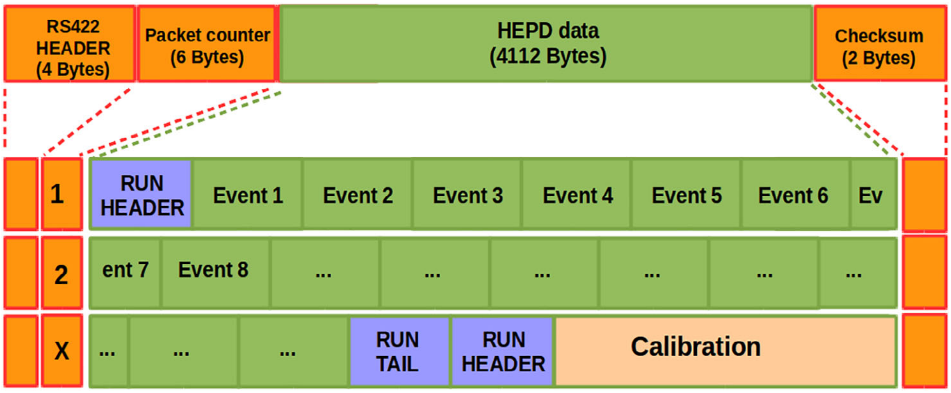


FIGURE 9 Example of the double encapsulation of HEPD scientific data. Each line represents an RS-422 frame of 4124 Bytes, containing only a portion of an acquisition/calibration session data [Color figure can be viewed at wileyonlinelibrary.com]

4.2.4 | Format of the run data file

The encapsulation provided by the download FPGA is used by the satellite to separate HEPD data from those of the other payloads. The RS-422 frame index is also used in the offline data analysis to reconstruct the correct sequence of frames in order to have a single output file for each acquisition session. This resulting file will contain all the information about the corresponding acquisition run (run type, run identification number, absolute time at the start/stop of the run, HEPD configuration, and subsystem status), as well as broadcast data from the satellite (GPS information, attitude parameters, and so forth) that are needed in the analysis to reconstruct the temporal information and the satellite coordinates of each triggered event. This information is given at the beginning and end of each run in header and tail packets, respectively.

Between header and tail, event data are formatted in event packets; every event packet includes a header (containing the run identification number, an event index, the event temporal tag, and the length of the event); data contents, which include compressed silicon data and PMT data; and a checksum (a simple algebraic sum of data content) for a total size of about 500 bytes per event. Figure 9 illustrates the double encapsulation of HEPD scientific data as they are stored in the satellite memory and downloaded to the ground stations.

5 | IN-FLIGHT SOFTWARE PERFORMANCE

HEPD was launched on board the CSES on February 2, 2018 from the Jiuquan Satellite Launch Center near Jiuquan, Gansu Province (China). The HEPD health check procedures started on February 6, 2018, by checking real-time telemetry data transmitted to the Xi’an Satellite Monitor and Control Center, were successfully run without showing any significant anomaly.

For safety reason, in the first phases after launch and during the commissioning phases, the HEPD was powered in SAFE mode (all electronics powered-on except for the HVPS) and after some checks, set in NOMINAL mode, enabling the HVPS that biases PMTs and silicon detector planes at operational HV values.

In the subsequent months, HEPD underwent the commissioning phase, ended on July 28, 2018. The HEPD commissioning activities were followed by an INFN team at the CSES Data Application Center located in a CEA-ICD Institute in Beijing.

Thanks to the high configurability of the detector by means of CAN bus telecommands, this phase was dedicated to the study of HEPD performances with different settings in order to define the optimal in-flight detector configuration. In particular, during the commissioning phase, different payload configurations were tested in order to define the optimal detector configurations. Several procedures were provided to satellite managers in order to set the required configurations, each procedure including the content and the detailed time schedule of the telecommands to be sent to HEPD, together with specific recovery actions. A correct time schedule is needed because telecommands sent from the ground station must be programmed to be sent while the satellite has “visual contact” with S band ground segment and so as not to interfere with the sequence of telecommands that the satellite automatically sends to the HEPD during normal functioning (e.g., the standby telecommand cyclically sent by the satellite before the passage in the polar regions). With these dedicated procedures, for example, it was possible to modify: the threshold values for the PMTs for the generation of the trigger signal, the values of the bias voltages for the PMTs, the system trigger configuration, or the definition of the orbital zones.

At the end of the commissioning phase, the optimal configuration was saved on the CPU FRAM (on both independent CPU sections), and HEPD was set in NOMINAL MODE and automatic operations.

In this state, HEPD performs automatically Acquisition Runs and Calibrations (except for the polar regions where it is set in the STANDBY MODE reduced power consumption regime) and no manual operations are needed.

At the present time, HEPD has collected data for more than 2 years and 14,000 orbits. Figure 10 shows the amount of HEPD raw data accumulated since the end of the commissioning phase. In the bottom panel, the number of discarded (because of a wrong checksum) RS-422 frames per day is shown. Since a frame contains just a few events, no attempt to recover corrupted frames is tried during the data analysis.

During these months of operation, some recovery actions proved necessary after the occurrence of some anomalous behaviors; the detailed analysis of collected telemetry data allowed to identify different kinds of anomalous behaviors, investigating the possible related cause and condition and performing the corresponding recovery actions.

In particular, from 28th of July 2018 to the end of December 2019 about a dozen anomalies in HEPD functionality was detected (see the vertical dashed red lines in Figure 11); most of them consists in the anomalous power off of the HEPD electronics due to a common cause affecting the low voltage power supply, and is strongly related to anomalous temperature increase detected on the CPU board. For this type of anomaly, HEPD functionality was automatically recovered by the CPU software which checks the other board status and, if necessary, powers on again the electronics components when exiting the standby zone.

A couple of anomalies concerned only the power off of the HVPS biasing the silicon detector planes and the PMTs, due to the intervention of the protection circuit. In these cases, for security purposes, the CPU software does not foresee any automatic recovery action; instead, it reports the HVPS error in telemetry data. The application of a power off and reboot sequence instructed by telecommands from the ground station recovered the normal functioning of the system.

Besides these minor anomalies, the CPU experienced a more severe kind of anomalous behavior that could be related to radiation damage. A single-event upset may be responsible of single/multiple bit changes causing a corruption of a local

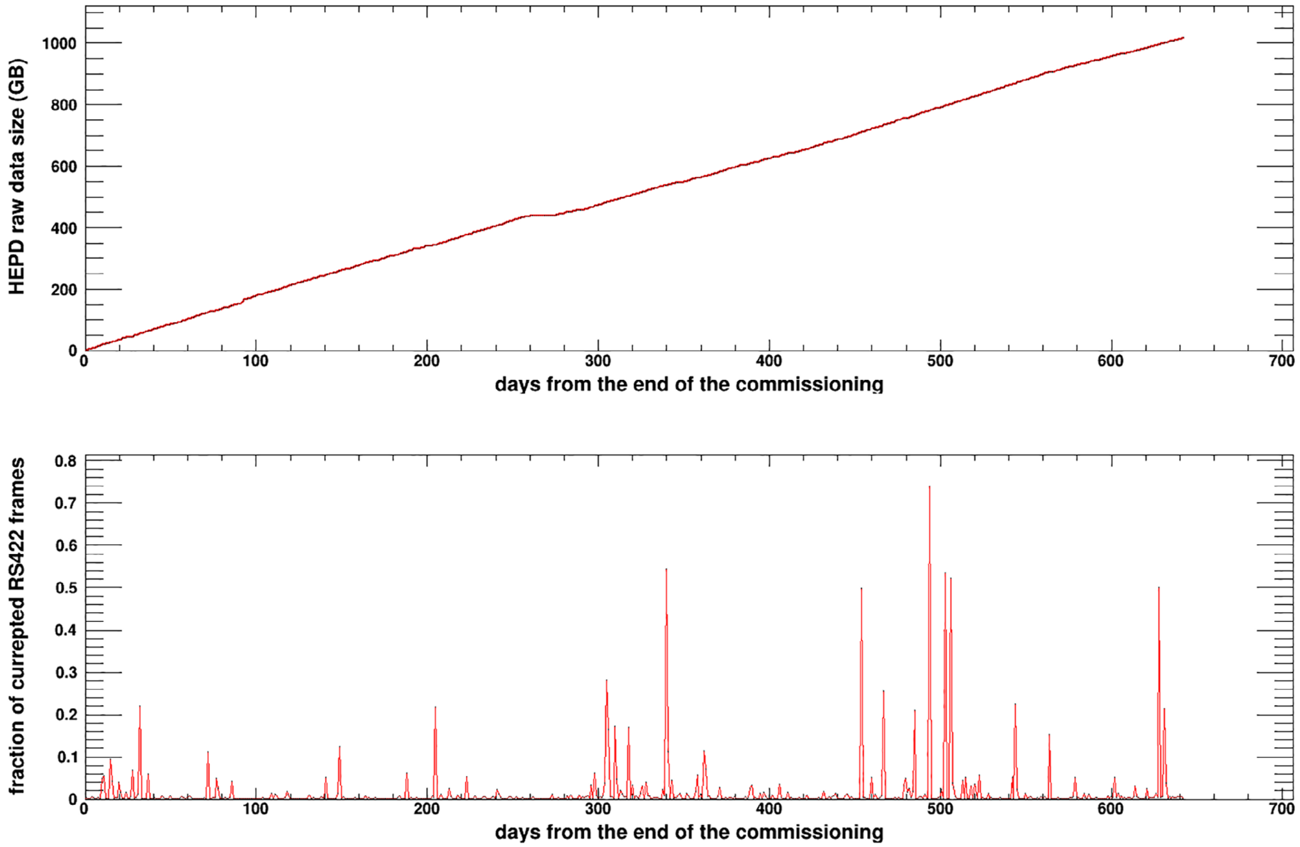


FIGURE 10 (Top panel) Raw data size collected from August 2018 (the end of the HEPD commissioning operations) to June 2020. (Bottom panel) Fraction of corrupted RS-422 frames per day in the same period [Color figure can be viewed at wileyonlinelibrary.com]

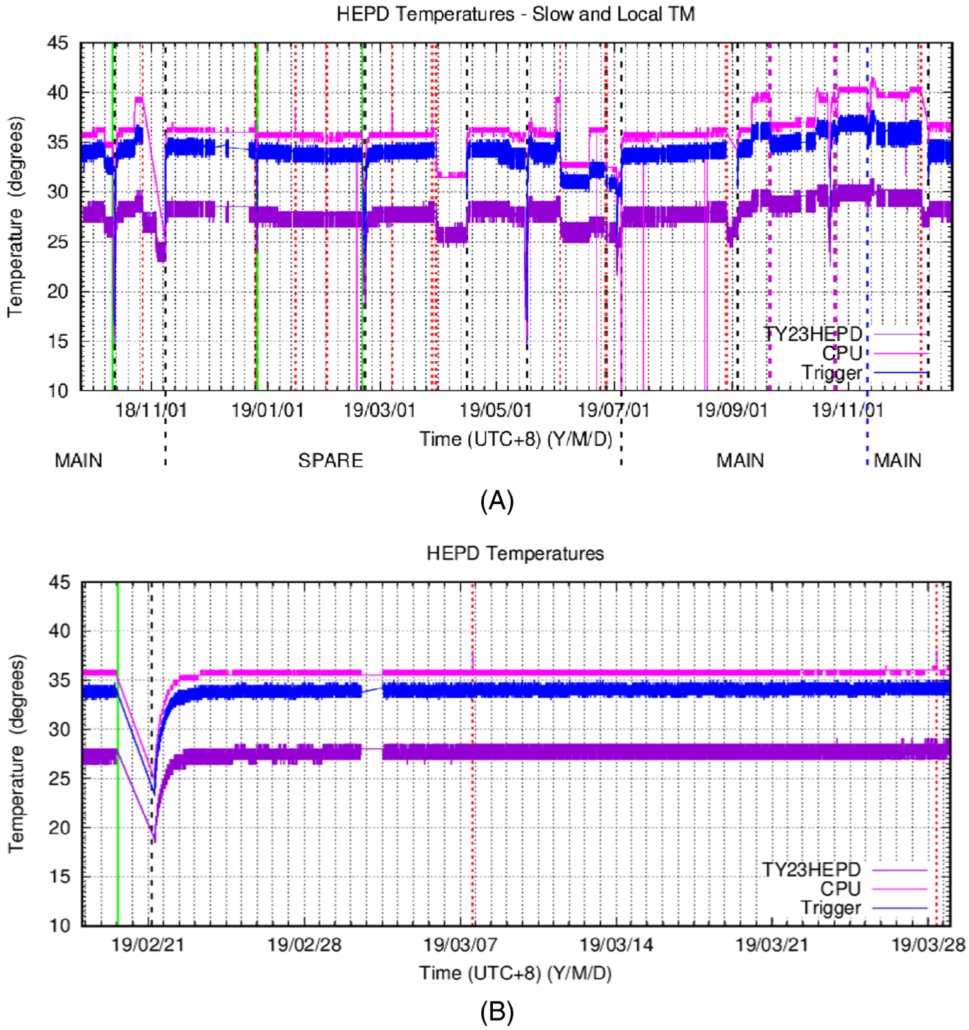


FIGURE 11 (A) Temperature as a function of time, read by the sensors placed on the CPU board (magenta), trigger board (blue), and HEPD baseplate (violet), as collected in HEPD telemetry data. Vertical dashed lines are related to the presence of an anomaly (red) or a planned system reboot for satellite maneuvers (black) or temperature increase (magenta). The vertical dashed blue line shows the application of the rewrite operation of the FRAM. Main and Spare refer to the two identical sections of the electronic boards, which are provided for redundancy purposes. (B) A zoom on the region including two anomalies. A spike in the CPU temperature can be noted at the same time of the anomaly [Color figure can be viewed at wileyonlinelibrary.com]

zone of memory (program or configuration). In particular, on June 3, 2019, a reboot signaled that the HEPD configuration, written on the FRAM of the CPU board, was corrupted. In this case, a dedicated procedure, implemented by a specific sequence of CAN bus telecommands, was successfully applied in order to rewrite the entire detector configuration stored in the FRAM, and restore the normal functioning.

6 | CONCLUSIONS

More than 2 years have passed since the launch of the CSES satellite and the HEPD detector is still working properly and acquiring high-quality data. The high number of exchangeable parameters of the DAQ software (as seen in Sections 4.2.1 and 4.2.2) allows changing the apparatus configuration by sending telecommands at any moment. Most of these parameters were appropriately configured during the intensive ground testing phase of the detector, but some small correction in the detector configuration was still necessary, even after the satellite launch, as expected. The separation of the software code in the application part and the configuration data (also written in different sections of the FRAM) allows to easily modify the latter by sending a few telecommands and minimizing the risks of compromising the operation of the system. In particular, after the major anomaly of November 2019 which compromised the functioning of the apparatus for a few days, it was necessary to completely rewrite the configuration data of the CPU FRAM in order to restore the correct system functioning. The rewriting procedure, finalized by sending several telecommands from ground, was ended successfully. The analysis of the telemetry data, provided by the continuous monitoring of the instrument, has also made it possible to identify the origin of most of the few anomalies found so far. The decision to automatically correct the minor anomalies while avoiding automatic operations for malfunctions affecting the HVPS system proved to be a safe choice. In conclusion, both CPU and DAQ software were found to be quite robust systems. The CSES-02 satellite, which is currently under development, will contain an improved version of the HEPD detector (named HEPD-02). Changes on the detector hardware will also require some changes in the software but apart from that, the software architecture used for HEPD-01 could be the basis for the new instrument, in order to obtain the same flexibility and reliability of the HEPD-01 software.

ACKNOWLEDGMENTS

This work was supported by the Italian Space Agency in the framework of the Accordo Attuativo No. 2016-16-H0 Progetto Limadou Fase E/Scienza (CUP F12F1600011005) and the ASI-INFN agreement n. 2014-037-R.0, addendum 2014-037-R-1-2017.

ORCID

Alessandro Sotgiu https://orcid.org/0000-0001-8835-2796
Cinzia De Donato https://orcid.org/0000-0002-9725-1281
Claudio Fornaro https://orcid.org/0000-0001-5542-8280
Simona Bartocci https://orcid.org/0000-0002-3066-8621
Roberto Battiston https://orcid.org/0000-0002-5808-7239
Donatella Campana https://orcid.org/0000-0003-1504-9707
Luca Carfora https://orcid.org/0000-0002-2341-9870
Guido Castellini https://orcid.org/0000-0002-0177-0643
Livio Conti https://orcid.org/0000-0003-2966-2000
Andrea Contin https://orcid.org/0000-0002-2535-5700
Fulvio De Persio https://orcid.org/0000-0003-4033-207X
Cristian De Santis https://orcid.org/0000-0002-7280-2446
Francesco M. Follega https://orcid.org/0000-0003-2317-9560
Roberto Iuppa https://orcid.org/0000-0001-5038-2762
Ignazio Lazzizzera https://orcid.org/0000-0001-5092-7531
Matteo Martucci https://orcid.org/0000-0002-3033-4824
Giuseppe Masciantonio https://orcid.org/0000-0002-8911-1561
Matteo Mergé https://orcid.org/0000-0002-2018-4236
Giuseppe Osteria https://orcid.org/0000-0002-9871-8103
Francesco Palma https://orcid.org/0000-0001-7076-8830
Federico Palmonari https://orcid.org/0000-0003-3707-0013
Alexandra Parmentier https://orcid.org/0000-0002-9073-3288
Francesco Perfetto https://orcid.org/0000-0001-8119-5046
Piergiorgio Picozza https://orcid.org/0000-0002-7986-3321
Mirko Piersanti https://orcid.org/0000-0001-5207-2944

REFERENCES

1. Shen X, Zhang X, Yuan S, et al. The state-of-the-art of the China Seismo-Electromagnetic Satellite mission. *Science China Technological Sciences*. 2018;61(5):634-642. http://dx.doi.org/10.1007/s11431-018-9242-0.
2. De Santis A, De Franceschi G, Spogli L, et al. Geospace perturbations induced by the Earth: The state of the art and future trends. *Physics and Chemistry of the Earth, Parts A/B/C*. 2015;85-86:17-33. http://dx.doi.org/10.1016/j.pce.2015.05.004.
3. Lin J, Shen X, Hu L, Wang L, Zhu F. CSES GNSS ionospheric inversion technique, validation and error analysis. *Science China Technological Sciences*. 2018;61(5):669-677. http://dx.doi.org/10.1007/s11431-018-9245-6.
4. Cao J, Zeng L, Zhan F, et al. The electromagnetic wave experiment for CSES mission: Search coil magnetometer. *Science China Technological Sciences*. 2018;61(5):653-658. http://dx.doi.org/10.1007/s11431-018-9241-7.
5. Cheng B, Zhou B, Magnes W, Lammegger R, Pollinger A. High precision magnetometer for geomagnetic exploration onboard of the China Seismo-Electromagnetic Satellite. *Science China Technological Sciences*. 2018;61(5):659-668. http://dx.doi.org/10.1007/s11431-018-9247-6.
6. Li XQ, Xu YB, An ZH, et al. The high-energy particle package onboard CSES. *Radiation Detection Technology and Methods*. 2019;3(3). http://dx.doi.org/10.1007/s41605-019-0101-7.
7. Ambrosi G, Bartocci S, Basara L, et al. The HEPD particle detector of the CSES satellite mission for investigating seismo-associated perturbations of the Van Allen belts. *Science China Technological Sciences*. 2018;61(5):643-652. http://dx.doi.org/10.1007/s11431-018-9234-9.
8. Aleksandrin SY, Galper AM, Grishantzeva LA, et al. High-energy charged particle bursts in the near-Earth space as earthquake precursors. *Annales Geophysicae*. 2003;21(2):597-602. http://dx.doi.org/10.5194/angeo-21-597-2003.

9. Sgrigna V, Carota L, Conti L, et al. Correlations between earthquakes and anomalous particle bursts from SAMPEX/PET satellite observations. *Journal of Atmospheric and Solar-Terrestrial Physics*. 2005;67(15):1448-1462. http://dx.doi.org/10.1016/j.jastp.2005.07.008.
10. Battiston R, Vitale V. First evidence for correlations between electron fluxes measured by NOAA-POES satellites and large seismic events. *Nuclear Physics B - Proceedings Supplements*. 2013;243-244:249-257. http://dx.doi.org/10.1016/j.nuclphysbps.2013.09.002.
11. Martucci M, Boezio M, Bravar U, et al. Analysis on H spectral shape during the early 2012 SEPs with the PAMELA experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. 2014;742:158-161. http://dx.doi.org/10.1016/j.nima.2013.11.078.
12. Martucci M, Munini R, Boezio M, et al. Proton fluxes measured by the PAMELA experiment from the minimum to the maximum solar activity for solar cycle 24. *The Astrophysical Journal*. 2018;854(1):L2. http://dx.doi.org/10.3847/2041-8213/aaa9b2.
13. Adriani O, Barbarino GC, Bazilevskaya GA, et al. TIME DEPENDENCE OF THE e−FLUX MEASURED BY PAMELA DURING THE 2006 JULY–2009 DECEMBER SOLAR MINIMUM. *The Astrophysical Journal*. 2015;810(2):142. http://dx.doi.org/10.1088/0004-637x/810/2/142.
14. Marcelli N, Boezio M, Lenni A, et al. Time dependence of the flux of helium nuclei in cosmic rays measured by the PAMELA experiment between 2006 July and 2009 December. *The Astrophysical Journal*. 2020;893(2):145. http://dx.doi.org/10.3847/1538-4357/ab80c2.
15. De Donato C, Martucci M, Sotgiu A, CSES-HEPD. First in-flight performances of the high energy particle detector on board CSES. Paper presented at: Proceedings of the 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). Manchester, UK; 2019:1-6.
16. Scotti V, Osteria G. The electronics of the HEPD of the CSES experiment. *Nuclear and Particle Physics Proceedings*. 2017;291-293:118-121. http://dx.doi.org/10.1016/j.nuclphysbps.2017.06.024.
17. Picozza P, Battiston R, Ambrosi G, et al. Scientific goals and in-orbit performance of the high-energy particle detector on board the CSES. *The Astrophysical Journal Supplement Series*. 2019;243(1):16. http://dx.doi.org/10.3847/1538-4365/ab276c.
18. ADSP-2189M Datasheet, rev. A., Analog Devices, Norwood, MA; 2020. https://www.analog.com/media/en/technical-documentation/data-sheets/ADSP-2189M.pdf. Accessed April 7, 2020.
19. Adriani O, Barbarino GC, Bazilevskaya GA, et al. The PAMELA Mission: Heralding a new era in precision cosmic ray physics. *Physics Reports*. 2014;544(4):323-370. https://doi.org/10.1016/j.physrep.2014.06.003.
20. Adriani O, Barbarino GC, Bazilevskaya GA, et al. Ten years of PAMELA in space. *Riv Nuovo Cim*. 2017;40:473.
21. Russo S, Barbarino GC, Boscherini M, et al. The TOF and trigger electronics of the PAMELA experiment for the search of primordial antimatter in space. *Frascati Phys Ser*. 2004;XXXVII:107-112.
22. Zhou L, An J. Design of a payload data handling system for satellites. Paper presented at: Proceedings of the 2013 3rd International Conference on Instrumentation, Measurement, Computer, Communication and Control; 2013; Shenyang.
23. Di Natale M, Zeng H, Giusto P, Ghosal A. *Understanding and using the controller area network communication protocol*. New York, NY: Springer-Verlag; 2012.
24. Parkes SM, Armbruster P. SpaceWire: a spacecraft onboard network for real-time communications. Paper presented at: Proceedings of the 14th IEEE-NPSS Real Time Conference; 2005:6-10; IEEE, Piscataway, NJ.
25. Fornaro C, Cafagna FS, Osteria G, Scotti V, Perfetto F, Conti L. The onboard software of the EUSO-SPB pathfinder experiment. *Software: Practice and Experience*. 2019;49(3):524-539. http://dx.doi.org/10.1002/spe.2655.
26. Roosta R. A comparison of radiation-hard and radiation-tolerant FPGAs for space applications, NASA Electronic Parts and Packaging (NEPP) Program JPL D-31228; 2004.
27. Tsakalakos T, Ovid'ko IA, Vasudevan A, Asuri K. *Nanostructures: synthesis, functional properties and application*. Vol 128. Berlin, Germany: Springer Science & Business Media; 2012.
28. Ambrosi G, Bartocci S, Basara L, et al. Beam test calibrations of the HEPD detector on board the China Seismo-Electromagnetic Satellite. *Nuclear instruments and methods in physics research section a: accelerators, spectrometers, detectors and associated equipment*. 2020;974:164170. http://dx.doi.org/10.1016/j.nima.2020.164170.

How to cite this article: Sotgiu A, De Donato C, Fornaro C, et al. Control and data acquisition software of the high-energy particle detector on board the China Seismo-Electromagnetic Satellite space mission. *Softw Pract Exper*. 2021;51:1459–1480. https://doi.org/10.1002/spe.2947