Available online at www.sciencedirect.com

SCIENCE DIRECT®

Advances in Space Research 37 (2006) 1857–1861

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

The PAMELA Storage and Control Unit

M. Casolino [a,*], F. Altamura [a], A. Basili [a], M.P. De Pascale [a], M. Minori [a], M. Nagni [a], P. Picozza [a], O. Adriani [b], P. Papini [b], P. Spillantini [b], G. Castellini [c], M. Boscherini [d], M. Boezio [e], G. Alfarano [f], F. Sebastiani [f], S. Tassa [f]

[a] *INFN, National Institute of Nuclear Physics, Structure of Rome II, and Physics Department, University of Rome II "Tor Vergata", Via Ricerca Scientifica 1, I-00133 Rome, Italy*

[b] *INFN, Structure of Florence, and Physics Department, University of Florence, I-50019 Sesto Fiorentino, Italy*

[c] *CNR–Istituto di Fisica Applicata "Nello Carrara", I-50127 Florence, Italy*

[d] *Universität Siegen, D-57068 Siegen, Germany*

[e] *INFN, Structure of Trieste, and Physics Department, University of Trieste, I-34147 Trieste, Italy*

[f] *Nergal s.r.l., I-00177 Rome, Italy*

Received 6 November 2004; received in revised form 1 June 2005; accepted 5 June 2005

Abstract

The PAMELA experiment aims to measure with great precision the antimatter present in our Galaxy in the form of high energy particles; in the same time it will measure the galactic, solar and trapped components of cosmic rays. The experiment will be housed on board a Russian Resurs-DK1 satellite and launched in the year 2005 to fly a 350–600 km orbit with an inclination of 70.4°. All operations of the instrument – including data storage – are handled by the PAMELA Storage and Control Unit (PSCU), which is divided in a Central Processing Unit (CPU) and a Mass Memory (MM). The CPU of the experiment is based on a ERC-32 architecture (a SPARC v7 implementation) running a real time operating system (RTEMS). The main purpose of the CPU is to handle slow control, acquire and store data on a 2 GB MM. Communications between PAMELA and the satellite are performed via a 1553B bus. Data acquisition from the sub-detectors (Time-of-Flight counter, Magnetic Spectrometer, Electromagnetic Calorimeter, Anticoincidence shield, Neutron Detector, and Bottom scintillator S4) is performed via a 2 MB/s interface. Download from the PAMELA MM towards the satellite main storage unit is handled by a 16 MB/s bus. The daily amount of data transmitted to ground has been evaluated in not more 20 GB. In this work, we describe the CPU of the experiment and the general software scheme.

Keywords: Cosmic rays; Antimatter; Satellite-borne experiment

1. Introduction

The scientific goals of PAMELA (Simon, 2003; Boezio et al., 2003; Sparvoli, 2004) include the accurate measurement of the antiproton and positron fluxes with a sensitivity, statistics and coverage higher than most previous (mainly balloon-borne) experiments. Other goals are the search for possible antinuclei, with a sensitivity better than 10^{-7} in the antihelium-to-helium ratio. For its characteristics PAMELA will also address several issues of solar and heliospheric physics (Casolino, 2004) in an energy range between 100 MeV and some hundreds of GeV. The PAMELA telescope includes a Scintillator/Time of Flight System (Barbarino et al., 2003), a Silicon Tungsten calorimeter (Boezio et al., 2002), a permanent magnet with a silicon microstrip tracker (Adriani et al., 2003), a Tail Scintillator, a Neutron

* Corresponding author. Tel.: +39 06 7259 4909; fax: +39 06 7259 4647.

E-mail addresses: Marco.Casolino@roma2.infn.it, casolino@roma2.infn.it (M. Casolino).

0273-1177/$30 © 2005 COSPAR. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.asr.2005.06.012

Detector (Galper et al., 2001) and an Anticoincidence System (Pearce, 2003; Orsi et al., 2004). It will be installed on board of the Russian Resurs DK-1 satellite and will be launched in the year 2005. The experiment is designed in an hierarchical/modular structure where the subdetectors handle all fast acquisition and data suppression before sending data to the acquisition board (IDAQ) that interfaces them to the CPU. Then data are stored into the Mass Memory (MM) before transfer to the satellite for downlink. Boards have redundancy and all detectors are divided in sections designed to continue working – although with degraded performances – even in case of several partial failures.

2. The PSCU

The PAMELA Storage and Control Unit (PSCU) (manufactured by Laben S.p.A.) handles all slow controls, the interactions between the satellite, and the data acquisition, storage and downlink. The PSCU box (see Fig. 1) is a space qualified system designed to be more fault tolerant than other systems and for this reason (as well as weight and power considerations) is not redundant. The PSCU is composed by:

- a central processor ERC32 SPARC v7 with a clock of 24 MHz employing a PROM of 128 kB for booting, 4 MB of RAM, and two 512 kB banks of EEPROM;
- an avionics standard 1553B BUS interface board from/to the Resurs satellite. The CPU executes all the commands received from the Resurs satellite through this bus; these commands can be used to set the acquisition modes, reprogram the subdetectors or the CPU itself. This interface also acts as a monitor used by the Resurs to asynchronously control that the PAMELA instrument is running properly;
- two mass memory modules of 2 GB each. These are used to store all data coming from the Front End (FE) boards, as well as the telemetry and housekeeping data. Latch-up detection is supported by using an Error Correction Code (ECC) employing 40 bits to store 32 bit words. When a latch-up occurs, the failing byte module is transparently switched to the safe one. In case of failure of one of the memory banks it is possible to switch to the second one;
- interface board (PAMELA InterFace – PIF). This high speed board:
  - transfers commands using the Command Interface (CMD/IF), and multiplexes them to different FE, using an Intermediate Data AcQuisition (IDAQ) board (see Fig. 2). Four pages of 16 kB each are employed for this purpose;
  - transfers physics data from the IDAQ to its internal SRAM (one page of 128 kB or two pages of 64 kB each);
  - transfers data from the SRAM into the MM;
  - transfers data from the MM into the Telemetry Adapter Module Interface (TAM/IF), connected with the satellite (see Fig. 2). When the data transfer to the satellite is ongoing as a burst of consecutive DMA programming, no other acquisition operation or writing into memory are in progress. Two hundred seconds are estimated to empty the MM in one single session.

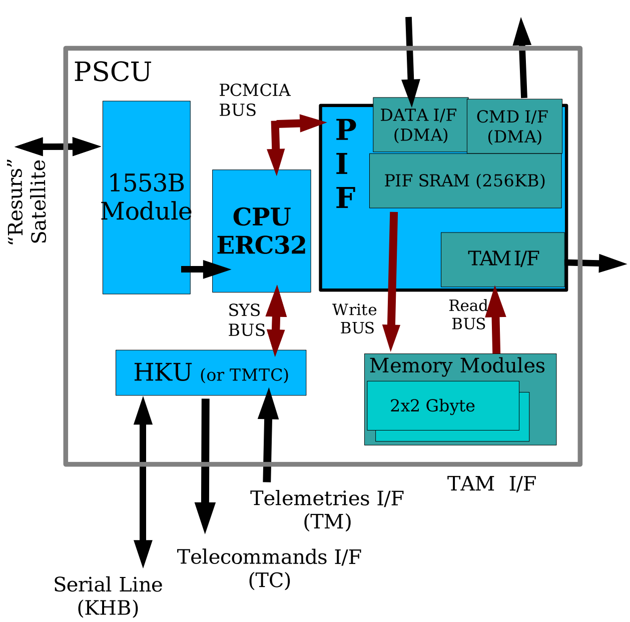

Fig. 1. Block scheme of the PAMELA Storage and Control Unit. See text for explanation of the various acronyms.

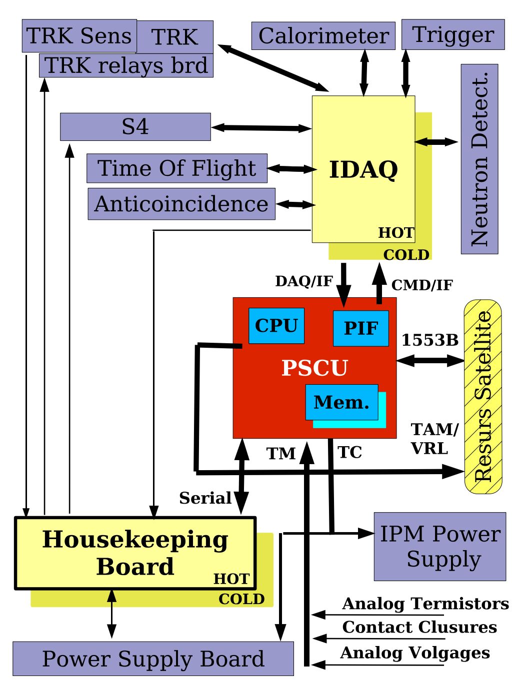

Fig. 2. Scheme of the PAMELA experiment and data flow and the relationship with the PSCU point of view. Redundancy of power supply is not shown in the picture.

- a multipurpose TeleMetry and Control board (TMTC) containing various digital, ADC/DAC interfaces to interact with the subdetectors and the subsystems of the experiment. A serial line is connected to a second Housekeeping Board (KHB) which handles reception of other engineering information and the issuing of slow control commands (see Fig. 2).

3. PAMELA acquisition mechanism and philosophy

The acquisition mechanism in PAMELA follows the following algorithm (Fig. 2):

(1) the PSCU programs the trigger board using a specific command to it via IDAQ CMD/IF. This prepares the system to receive an event;
(2) when the event (a particle crossing the instrument field of view) arrives, all FEs receive a trigger signal and execute the readout within their own subsystems;
(3) the PSCU sends to each FE, via the IDAQ board, a command to read the acquired data;
(4) the IDAQ demuxes data into the CPU through the Digital Acquisition Interface (DAQ/IF);
(5) the PSCU receives event data and stores them in MM;
(6) the cycle is repeated until some error occurs or the programmed duration of the acquisition loop has expired.

4. PSCU Software

The SPARC/ERC32 CPU is running RTEMS 4.0, a hard real time operating system, with microkernel model, that provides all primitives for multitasking, interrupt manager, timers, events, mailboxes, semaphores, signals, and clock handling (RTEMS). The PAMELA software application is running on top of the RTEMS system. The software handles communication from and to peripherals event acquisition. In addition, it sends FE commands, waits for the answers, writes them to the MM, and performs some default checks. This is what is typically done during the Initialization (which including FE DSP program loading), Calibration and Acquisition phases. The software also manages change of operative modes of peripherals according to orbital position (given periodically by the satellite via the 1553 bus). Beside, the PSCU software handles the programming of all subsystems and peripheral via a dedicated board (KHB). This includes slow control of Relays Board (to power on and off the secondary power supplies); the acquisition boards, etc. The software thus checks and handles the power management: turns on and off power supplies of all boards and FE, manage hot/cold switching and checks for anomalies. These operations are performed using specific commands to the Power Supply Board and high level commands. Telemetries (temperature, power, status of the power relays) coming from the Telemetry Interface are also periodically sampled and stored into the MM – for offline analysis on the ground (Nagni et al., 2004) – and into the telemetry format region of the 1553 interface. Besides, the PSCU also checks telemetry values in order to point out anomalies in performing certain actions.

All the software, including the RTEMS operating system, is written in the C programming language, cross-compiled under an ordinary Linux/Intel PC using the "sparc-rtems" porting of the GNU compiler and GNU tools (like gdb, ld, mn, etc.). Remote debugging, loading and firmwaring could be made using gdb. Many other utilities and languages have been used in the development: Perl, C++, and CVS.

The PAMELA software is based upon on a multitask architecture. It largely employs the communication and synchronization features provided by the operating system. Data transfer and synchronization between cooperating tasks and/or Interrupt Service Requests (ISRs) employes RTEMS tools like semaphores, events and message queues. The PAMELA application can be divided into three logical layers:

(1) Drivers and Modules layer: this layer is in charge of managing the protocol of boards, the interrupts, communication bus 1553B, MM drivers, File System, etc.;
(2) Low-Level Tasks layer: these tasks handle the high priority process like dispatching of the macrocommand and the accessing to shared resources like the TMTC board;
(3) High-Level (or Application) layer: these tasks handle the main flow of the application: manage the acquisition and FE's configuration, error and alarm conditions, perform power supplies management.

4.1. Tasks

As already mentioned, the PAMELA software is based upon a multitask architecture: operating system objects are all created at the software initialization (just after bootstrap) and never destroyed (Straumann, 2001). For each task, the system creates the task objects with its own priority and its entry point (a function pointer) and the corresponding mailbox for this task that implements a message queue. A message queue (FIFO) allows the message passing among tasks and ISRs and it can contain a variable number of messages.

In this work, we will describe the application level of the software only, ignoring many low-level implementation details.

Three high-level tasks manage the main application process:

- `PamManager`: receives most of the macrocommands from the `MCMDDispatcher` and manages them. This is the main state machine of the PAMELA software which controls the behavior of the other two tasks: `RunManager` and `SlowControlManager`. As soon as one of these task has completed the requested operation it notifies to the `PamManager` the end of the operations with a return code and an optional error code. Management of all macrocommands is also done by this task;
- `RunManager`: it is the task for the acquisition process. Its goal is to assemble command queues for different FEs, store them into the PIF, send them to the IDAQ, wait for the answer, store results into the MM, detect some FE error condition and locally manage some of them;
- `SlowControlManager`: this task manages all monitoring signals, power supply, temperatures, voltages, via the dedicated boards (TMTC and KHB). Furthermore it checks and sets the correct power supply configuration after the boot process of PAMELA, handles the power on and power off of PAMELA and of its FE and manages recovery and error handling in case of anomaly.

4.2. Acquisition software procedure

The acquisition procedure is performed by the dedicated `RunManager` task. This task is able to be asynchronously stopped, cleared and restarted. The application software holds a set of parameters, including commands to be sent to different FE. Some of them are dynamically managed and they can be changed from ground using a specific macrocommand. Some others are statically linked to a data structure inside the firmware of the program. The `RunManager` performs the following main operations:

- Initialization: a set of commands is sent to different FE in order to be initialized. Tracker, Calorimeter, Trigger, Time-of-Flight, Anticoincidence and S4 need to be initialized. Tracker, Anticoincidence and Calorimeter also need the DSP program to be loaded. The initialization procedure depends on many PAMELA parameters and internal states of the software that can be changed from ground. The procedure is frequently repeated up to once per orbit ($\sim$90 min);
- Calibration: a set of commands are sent to different FE in order to calibrate them. Tracker, Calorimeter, S4 and Anticoincidence all need to be calibrated. The calibration procedure for a single FE is typically a series of commands. The procedure is frequently repeated up to once per orbit;
- Acquisition Loop (or Run): this is done in two steps. First a series of commands is sent to each FE for acquisition initialization, then the Acquisition command Queue (AQ) is assembled and stored in a command page of the PIF. Second, the software enters in a loop during which it continuously:

(1) sends the AQ to the IDAQ via CMD/IF;
(2) waits for the answer from the DAQ/IF;
(3) writes the data into the MM.

The Initialization is done after the PSCU boot, before Calibration and after a system reset. A system and FE reset is performed via the KHB. A system reset can be performed in case of alarms detected by the housekeeping board, upon request by error recovery procedures and so on. The Calibration is performed after each ascending node[1] (or a multiple of the ascending node). Following the orbit of the Resurs, the satellite sends to PAMELA a special macrocommand named CALIBRATE every time the ascending node of the orbit is reached. Acquisition is started immediately after an initialization or after a calibration procedure. The duration of the Run loop depends on the Working Schedule mode of PAMELA (WS) as explained in Section 4.3.

4.3. Acquisition modes and working schedule modes

Data acquisition of Pamela is limited by the amount of information that can be transmitted to the ground. Particle rate in Low Earth Orbit depends from the position of the satellite, generally increasing toward the poles (due to the lower geomagnetic shielding). There is, however, the exception of the Brazilian Anomaly – a region situated in the South Atlantic – where the spacecraft crossed the trapped protons composing the inner radiation belt. In this region particle rate can increase by several orders of magnitude, resulting in a fast filling of the memory. The acquisition of PAMELA has thus been designed to avoid this potential problem; it can basically operate according to two different modes depending of the Trigger Board settings.[2]

(A) in this mode the Trigger Board selects a good event getting a signal from all three ToF scintillator

[1] The ascending node is the point where the orbit of satellite crosses the plane of Earth's equator going towards the North. This is performed to avoid the South Atlantic Anomaly (see the following section).

[2] There are many more possible trigger schemes: in here we discuss only the main two for sake of simplicity.

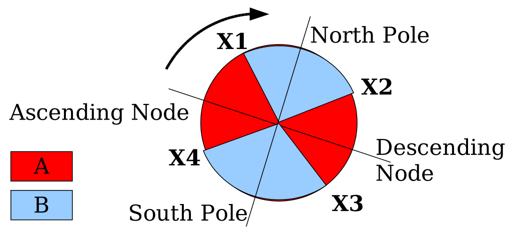

Fig. 3. The Working Schedule 1 orbit mode. In this mode, the standard acquisition (A – S1*S2*S3) is performed at low latitude regions, whereas the more conservative trigger (B – S2*S3) is used in the high latitude regions to avoid a high trigger rate during passage through the electron radiation belts. Note that this mode does not take into account the passage in the Brazilian Anomaly.

planes: S1 on the top of the payload, S2 and S3, respectively, above and below the spectrometer;

(B) in this mode, designed for the passage through the radiation belts, the Trigger Board employs only the S2 and S3 planes. This mode is also known as *safe mode* for the PAMELA acquisition.

Every time an acquisition procedure is started, the software selects the mode A or B and the duration of the Run loop. These two data are derived depending on the WS mode of PAMELA. This is a parameter of PAMELA that can take four different values:

0. *Safe mode*. The acquisition mode is always in B. The duration of the run is a predefined value (30 min) that can be changed from ground;
1. *Automatic mode*. Thanks to the CALIBRATE macrocommand coming at the ascending node (see Section 4.2), the software knows the start of the orbit and, from to the internal clock of the PSCU, can estimate the position of the satellite. In this conditions, the software is able to schedule the duration of the run following the scheme shown in Fig. 3. X1, X2, X3 and X4 are predefined values in seconds starting from the ascending node. They can be changed from ground and their default values are, respectively, 720, 2160, 2700 and 5490 s. These values ensure that PAMELA will work in a *safe mode* near the high latitude radiation belts but does not take into account the passage in the South Atlantic Anomaly (see the following item);
2. *Custom mode*. A table is uploaded from ground via macrocommand after time synchronization with the satellite is performed. This table stores the absolute values of Moscow time and modes (0 or 1). Sixteen tables can be uploaded and stored in the EEPROM, for a total coverage of dozens of days.[3] In this way, it is possible to take into account from ground of all passages in the radiation belts (high latitude and the South Atlantic Anomaly);
3. *S1 rate meter* of PAMELA. The application software selects the appropriated working mode reading periodically the rate meter of the scintillator plane S1[4]: if this value is over a predefined threshold the software sets the acquisition mode to *safe mode*; then, if this value goes below another predefined threshold the mode is set to 0.

4.4. Conclusions

In this work, we have discussed the functionalities of the PAMELA Storage and Control Unit. All tests performed up to now have shown the correct behavior of the device and its interaction with the detectors. Also the interface with the satellite has been verified on the technological model. Integration in the flight model is expected to occur in Summer 2005; the launch will take place from the cosmodrome of Baikonur at the end of 2005.

References

Adriani, O. et al. The magnetic spectrometer of the PAMELA satellite experiment. Nucl. Instr. Meth. Phys. Res. A 511, 72, 2003.

Barbarino, G.C. et al. The PAMELA time-of-flight system: status report. Nucl. Phys. (Proc. Suppl.) B 125, 298, 2003.

Boezio, M. et al. A high granularity imaging calorimeter for cosmic-ray physics. Nucl. Instr. Meth. Phys. Res. A 487, 407, 2002.

Boezio, M. et al. The space experiment Pamela. Nuc. Phys. B (Proc. Suppl.) 134, 39, 2003.

Casolino, M. Cosmic ray observation of the heliosphere with the PAMELA experiment, these proceedings, 2004.

Galper, A.M., et al. Measurement of primary protons and electrons in the energy range of 10^{11}–10^{13} eV in the PAMELA experiment, in: Proceedings of the 27th ICRC, Hamburg, OG 2219, 2001.

Casolino, M., De Pascale, M.P., Nagni, M., Picozza, P. YODA++: A proposal for a semiautomatic space mission control, these proceedings, 2004.

Orsi, S., et al., these proceedings, 2004.

Pearce, M. The anticounter system of the PAMELA space experiment, in: Proceedings of the 28th ICRC July 31–August 7, Tsukuba, Japan, OG 1.5, 2125, 2003.

RTEMS official documentation http://www.rtems.com/onlinedocs/releases/4.0.0/doc.

Simon, M. Status of the PAMELA experiment on-board of the Resurs DK-1 spacecraft, in: Proceedings of the 28th ICRC July 31–August 7, Tsukuba, Japan, OG 1.5, 2117, 2003.

Sparvoli. R., et al., these proceedings, 2004.

Straumann, T. Open source real-time operating systems overview, in: Proceedings of the 8th International Conference on Accelerator and Large Experimental Physics Control Systems, San Jose, USA, 2001.

[3] Particle rate can also increase due to transient effects due to Solar Particle Events: in this case a specific working table has to be sent from ground.

[4] This is performed by reading one event every 8 s from the PIF into the CPU instead of sending it directly to the MM.