The Data Processor System of EUSO-SPB

G.Osteria[*] **, F. Perfetto, V. Scotti**

Istituto Nazionale di Fisica Nucleare - Sezione di Napoli, Italy
E-mail: osteria@na.infn.it

F. Cafagna

Istituto Nazionale di Fisica Nucleare - Sezione di Bari, Italy

C. Fornaro

Università Telematica Internazionale Uninettuno, Dipartimento di Ingegneria, Rome, Italy

for the JEM-EUSO Collaboration

We will present the Data Processor (DP) and the flight software of EUSO-SPB. The EUSO-SPB experiment is a pathfinder mission of the EUSO program having as main objective to detect Ultra-High Energy Cosmic Rays (UHECR) by measuring, for the first time from the space, the fluorescence and the Cerenkov light produced by the interaction of the particle with the nuclei of the Earth's atmosphere. The instrument is a telescope of smaller dimension in respect to the one designed for the ISS, mounted on an unpressurised gondola of a super pressure pumpkin balloon. EUSO-SPB was launched from the Columbia Scientific Balloon Facility of Wanaka Airport (New Zealand) on the 24th of April 2017, 22:50 and lasted for 12 days 4.5 hours aloft.

---

[*]Speaker.

1. Introduction

The EUSO-SPB experiment is a pathfinder mission of the EUSO program having as main objective to detect Ultra-High Energy Cosmic Rays (UHECR) by measuring, for the first time from near-space, the fluorescence and the Cerenkov light produced by the interaction of the particles with the nuclei of the Earth's atmosphere. The EUSO-SPB instrument [1] consists of a Fresnel optics made from 2 square UV transmitting PMMA (polymethyl methacrylate) lenses, a focal plane detector, Photo-Detector Module (PDM), equipped with readout electronics and powered by a High Voltage system and a Data Processor (DP). The instrument was equipped with two pre-calibrated (NIST) photodiodes, to measure the intensity of the light reaching the focal surface (day/dark detector), and with a differential GPS receiver (Compass) to measure its orientation during the flight. In addition two auxiliary instruments were used in combination with the main camera. An infrared camera (UCIRC) [2], to monitor the clouds coverage along the trajectory of the flight, and the Silicon photomultiplier Elementary Cell Add-on camera (SiECA) [3] an additional piece of focal surface, mounted adjacent to the PDM, realized by using SiPM's as photo-detectors. EUSO-SPB was launched on April the 24th 2017 from the Wanaka airport (New Zealand). The flight lasted about 12 days. A full description of the mission is given in [4]. Here, we describe both the concept and the performance during the flight of the data processor system which manages the main instruments and all the equipments on board while satisfying the severe restrictions imposed by a balloon-borne observation.
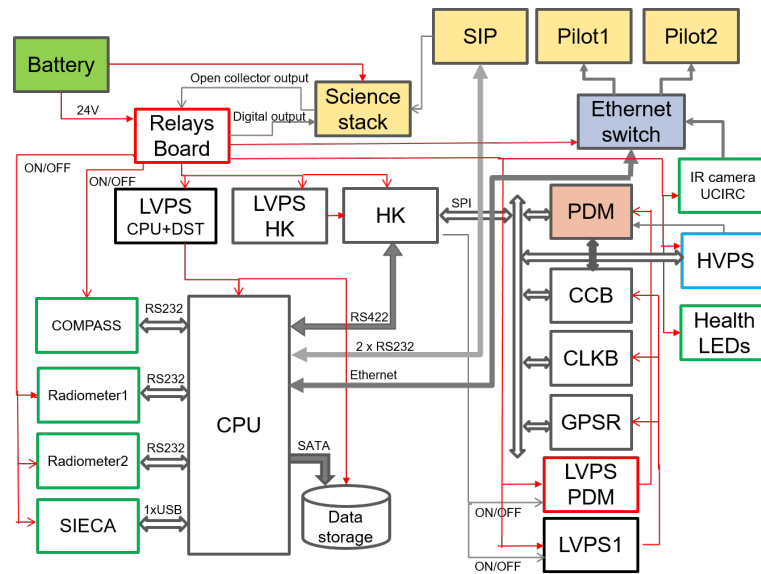
2. The Data Processor

The DP of EUSO SPB instrument is a complex system performing different tasks. Several modules have been developed in order to implement the functionalities required to control and monitor the instrument, during commissioning on ground and in flight. The main interface of the DP with the PDM front-end electronics is performed by the Control Cluster Board (CCB) [5]. The tagging of an event with arrival time and payload position as well as the time synchronization is performed by the Clock Board (CLKB) [6]. The monitoring and the control of the Voltage Power Supplies (LVPS and HVPS) as well as the acquisition of the temperature sensors is in charge of the House Keeping (HK) board [7]. The Central Process Unit (CPU) module is the interface of the DP with the NASA telemetry system (SIP and PILOTs) either via two serial ports and via an Ethernet switch, with the infra-red camera as well as all other equipments present on board.

The DP modules are hosted, together with the low voltage power supplies (LVPS) modules, in the DP mechanical sub-rack (Eurocard double size sub-rack). A block diagram of the DP is shown in Figure 1.

2.1 GPS receivers

Two GPS receivers (Oncore M12+ manufactured by Motorola) working in parallel for redundancy, are hosted in the GPSR mudule of the DP system. Both receivers are interfaced to the DP through the CLK-Board which associates the position, velocity, time, and satellite tracking status information measured by the GPS receivers to each acquired event. The time synchronization is ensured by the 1 Pulse Per Second (1 PPS) outputs of the receivers which are properly managed

Figure 1: Block diagram of the Data Processor.

by the CLK-Board; the data and the commands are transmitted through the RS232 communication ports. The receiver are capable of tracking twelve satellites simultaneously. To comply with International Traffic in Arms Regulations (ITAR) restrictions, there is a limit in velocity (515 m/s above 18 km altitude). The working temperature range is -40°C + 85°C.

2.2 The Clock board

The CLK-Board hosts the interface with the GPS receiver, it tags the events with their arrival time (UTC) and payload position. It generates the main clock (40 MHz) and the GTU clock (400 kHz) which are distributed to all the other subsystems. The CLKB receives first and second level trigger signals from the PDM or CCB respectively and provides the Veto and Busy signals to properly synchronize the data acquisition process. The CLK board can generate a trigger signal on a CPU command or in coincidence with the 1 PPS signal provided by the GPS receivers or at a fixed frequency of 20 Hz. It also measures the operating time and dead-time of the instrument and provides signals for time synchronization of the event. Finally the CLKB provides LVDS synchronizing 1PPS and 40MHz clocks as well as the external trigger to the SiECA camera.

Most of the functions of the CLKB are implemented in a FPGA Xilinx Virtex-5 XC5VLX50T.

2.3 The Cluster Control Board

The CCB [5] has a direct connection to the PDM board through a 40 MHz parallel bus. It processes and classifies the received data to perform the 2nd level trigger. In addition, CCB passes the clock signals from the CLK-Board to PDM board and the configuration data from CPU to PDM. The circuit is developed around a Xilinx Virtex-4 FX-60 with extended industrial temperature range.

2.4 CPU and Data Storage

The CPU module is a low power PCI Express embedded controller. It is based on an AMD Fusion G-Series APU high performance 1.0 GHz Dual-Core processor with a DDR3 memory interface operating at up to 800 MT/s. A Serial-ATA (SATA) controller provides a fast 3.0 Gbit/sec connection to the hard drives. The CPU collects data from the instruments through a four channels PCI/ SpaceWire board. Two SpaceWire links (200 Mbits/sec) are connected to CCB and CLKB modules for initialization and data storage and transmission. all data are stored on board using two 1 GB Solid-State Drives (SSD) operating in fault-tolerant RAID-1 configuration disks (Redundant Array of Independent Disks). The working temperature range of the two hard-disk is -40°C + 85°C.

2.5 HK system

The House-Keeping system [7] collects telemetry from the sub-systems of the instrument in slow control mode. It is responsible for monitoring voltages and currents of the Low Voltage Power Supply, and actuates power switch of several sub systems. It uses a serial bus to convey telemetry and telecommands through the CPU interface and to other sub-systems. The HK is implemented around an off-the-shelf micro-controller board (Arduino Mega 2560), combined with 5 custom-made protocol interface boards to pre-process the various signals.

3. The DP interface with the NASA telemetry system

The remote control of the balloon is based on the Support Instrumentation Package (SIP). The SIP, built by the Colombia Scientific Balloon Facility, houses power, flight computers, telemetry systems, and the communication systems. Every SIP has a similar architecture based on COMM1 and COMM2 on board computers, implementing a redundant flight telemetry system. The DP is connected to both COMM1 and COMM2 low-rate science ports through two RS232 serial ports. Both science return telemetry and forward command data are broadcast via these interfaces. For this flight the actual telemetry link could be either an IRIDIUM or Radio based one, depending on the launch phase. Along with the standard SIP's telemetry links, also two Iridium PILOT devices where installed in the SIP, featuring the Iridium OpenPort system, based on Ethernet interface implementing both TCP/IP and UDP based data flow. Thanks to the higher bandwidth, PILOTs have been preferred to transmit both housekeeping telemetry and science data to ground; at the same time have been used as a backup link for the instrument control commands and expert intervention on the DP-CPU. The average observed throughput is between 70 and 100 Kbps. The SIP makes available a science stack which provides analog and digital return telemetry and discrete command outputs. The science stack has been used to acquires several temperature sensors (analog return telemetry), to check the status of the solar panels system (digital return telemetry) and to switch ON/OFF, through the relays box, the power supplies of the all equipments present on board (discrete commands outputs).

4. The DP control and acquisition software

4.1 The DP Control Software

The DP plays the fundamental role of being the interface between detector and telemetry

blocks, and the end-users. For this reasons the control software has been designed to be flexible enough to be used in every phase of the detector commissioning and flight, guaranteeing the full access, control and monitor of the detector, as well as the storing of all relevant data, i.e. scientific, housekeeping and logging ones. Moreover, by design, the full user control via network, telemetry or console interfaces has been enabled, as well as a quick and comprehensive user configuration. Moreover the control software is modular by design, to cope with different DP configurations during test, commissioning and flight phases or to handle any potential block failure during the flight. Least but not last, the control software should permit development, test and detector control from different geographical location.

The latest requirement has been satisfied implementing a client server architecture on an highly configurable message passing tool, enabling any computer in the network, to send commands to the DP.

The DP control software can be subdivided into three layers implementing different tasks and responsibilities. A first layer implements the user and telemetry interfaces; a second one implements all the managers that steer and control the run processes, configurations and command exchanges; a third one implements all the low level tasks and interfaces with the relevant DP subsystems. The first two levels deal with higher level processes, implementing asynchronous communication via Ethernet, on both TCP/IP and UDP, and serial lines, as well as message passing using shareable memories. The third layer includes lower level processes and drivers implementing the actual interface to a given subsystem.

Every process, controlling a subsystem, implements asynchronous interprocess communication using shareable memory and conditional variables, along with configurable sequence, sequencer, consumer and logger managers. A sequence of operations can be stored or executed by the sequence manager. Stored sequences can be executed by the sequencer manager repeating them a given amount of time, or with a given frequency. Data produced by a subsystem, can be consumed by the consumer manager; it can add optional header and trailer blocks composed by a consumer manager ID, timestamp or a 16 bit CRC and pass the whole block to a collection of consumers. Several consumers can be defined, namely: UDP and TCP/IP network broadcasters, binary or text file dumpers, shareable memory broadcasters, a standard output printer and statistical collectors and retrievers. The logger manager handles a collections of standard output duplicators which copies every record to a file or broadcast it to a given IP via UDP or TCP/IP. Every manager is user configurable so that the process behavior can be customized according to needs and operational phases. Least, but not last, every process has user defined spawn capabilities, i.e. can spawn any command to the operating system via a user command issued by the client.

Besides the above described common features, specialized object implements the interface with a given subsystem. In particular: asynchronous communication via serial port, implements the housekeeping board and the radiometer interfaces; TCP/IP client implements the infra-red camera interface; TCP/IP listener implements the compass interface; a POSIX message queue asynchronous transmitter and receiver implements the interface with the science data collector program. Moreover two different interfaces have been written for the server processes: a TCP/IP and a serial one. The first to listen to any request from the network, the latter from the NASA telemetry system.

Thanks to the client-server architecture, any computer on the network or in the NASA teleme-

5

try system, can send a command to be executed on any DP-CPU subsystem, featuring an highly configurable message passing tool. In the flight configuration three servers where listening for commands to be routed to seven processes controlling the communication with as much as subsystems. A total of more than 800 subsystem commands, slightly less than 500 process commands and more than 700 sequences where defined and made available for the clients used by the payload operators and experts.

4.2 The DP data acquisition software

data handler is the EUSO-SPB application used for science data acquisition from subsystems: CLKB and CCB-PDM. It can also manage SiECA, if enabled. It essentially gets and saves all the data exchanged between CPU and CLKB, and between CPU and CCB/PDM. To do that, *data handler* performs many other operations: the initialization and setting of these subsystems, monitoring the SpaceWire connections used to connect them to the CPU, checking some parts of subsystems behavior (main control is performed by the housekeeping program suite), responding to external commands from the user or the housekeeping programs, performing S-curves, spawning and managing asynchronous SpaceWire buffer, etc. It can also be used to initialize SiECA and trigger it. *data handler* can be instructed to perform many types of acquisitions, among them: getting external events (cosmic rays or CLKB/GPS pulses), providing internal simulated events (CPU triggering), performing S-curves. In automatic cooperation with the dump program event dump can extract the essence data of single acquisitions or an entire S-curve. *data handler* can be launched and monitored as an interactive program by an operator and/or by another program (the housekeeping controller). To this purpose, different ways of interaction are provided: via console commands and via Message Queues. This allows also requesting *data handler* the status of the acquisition and of the subsystems, and changing some parameters without being restarted. Message Queues are the main command interface to the housekeeping applications. Once started, *data handler* outputs its activity log to console or to the activity log file. Data exchanged between CPU and CLKB/CCB/PDM are saved to the packet log file, enveloped in cpu data packets. A cpu data packet is composed of a header, the actual data, and a CRC32 for error protection (the envelope amounts to 22 bytes). There are of 4 types of cpu data packets:

1. commands from CPU to CLKB (3 bytes) or to CCB/PDM (13 bytes)

2. responses from CLKB and CCB to the CPU

3. science data:

    (a) L1 trigger from CCB (13 bytes)

    (b) L2 trigger from CCB (13 bytes)

    (c) CLKB event part from CCB (information produced by CLKB after a full event occurred, it contains: position and time (GPS), GTU timestamp, event counter, live and dead time, other, 1049 bytes)

    (d) CCB event part from CCB (information produced by the CCB after a full event occurred, it contains data produced by the PDM: GTU timestamp, trigger counter, PC counters for each GTU, other, 341019 bytes)

6

4. information produced by *data handler* itself (timestamps, command line parameters, *data handler* versions, etc.), size varies with content (500 bytes).

CLKB and CCB event parts come separated from the two homonym subsystems; their sequence counters are compared to verify they belong to the same event (CLKB and CCB sequence counters are zeroed at the beginning of each run). If the two parts belong to the same event, they are saved to the packet log file as a unique data packet. In case the two parts are not related, the older one is discarded and logged. Unrelated parts are never combined in case of temporary subsystem failure. For performance and to separate CPU I/O requests from SpaceWire I/O operations and timings, data exchanges over SpaceWire connections use an asynchronous buffer (spwd) and data packets are stored in shared memory: *data handler* will directly access and use them without any need to make time-consuming copies. All the blocking operations have a timeout to avoid blocking the process indefinitely in case of long or permanent failure. After a predefined number of timeouts or errors (calculated in order to have about 15 seconds of grace period) *data handler* stops, reporting the reason on the screen or the log file, and sending a high priority message to the output message queue. Other types of blocking problem (e.g. on SpaceWire) are recognized and recovered by *data handler* itself.

5. The flight

On April 24, 2017 at 22:51 Universal Time (UT)) EUSO-Balloon was launched successfully from Wanaka Airport (New Zealand). A leak in the balloon was confirmed on its third day of flight. The balloon started experiencing significant altitude drops at night when the temperature dropped, regaining its predicted altitude during the day as the temperature rose. The flight was terminated at 11:24 p.m. EDT, Saturday, May 6, after 12 days, 4 hours and 34 minutes aloft.

5.1 Instrument control during flight

The data acquired on board was transmitted through the two IRIDIUM PILOT systems to the Ground Support Equipment at the CSBF facility in Palestine, Texas. From there, the data was sent to a server based at the Colorado School of Mines. The flight was followed from several command centers located in USA, Japan and Europe.

5.2 Data acquisition during the flight

The instrument was launched with all the subsystems powered but the High Voltage of PMT's because it was not yet dark. During the flight the status of the apparatus was continuously checked by monitoring the HK telemetry data send from the paylod to ground once per minute together with the GPS coordinates and radiometers' data. The last two informations were used to verify the condition to start the measuremt (dark time) or stop the measurement (day time). Four acquisition modes were defined:

- Dark time: data taking

- Day time: data transmission

- Day to night transition: check the light before switching ON

• Night to day transition: switch OFF the HV

Acquisition modes were changeable from ground.

6. Data Processor performance

During the 12 days and 4 hours of operation the DP system was continuously up taking data during the dark periods and transmitting data to the ground (after compression) 24 hours a day. The CPU worked all the time without any flaw and never needed a reboot. The system acquired data for a total of about 40 hours, 60 GB of data was recorded on disk. Due to the relatively slow throughput of the telemetry system only 40 GB of the acquired data were transmitted to the ground together with telemetry data (voltages, currents, temperatures, NIST, Compass) and data from UCIRC and SiECA cameras before the splash down.

7. Conclusions

The EUSO SPB flight has demonstrated the maturity of the key technologies and methods that will be used for JEM EUSO. Data Processor system successfully provided the full control of the instrument and smooth data taking for the full duration of the flight. All the subsystems of the DP worked as intended.

References

[1] S. Bacholle et al. fer the JEM-EUSO Collaboration, "The EUSO-SPB Instrument" *Proceedings of the 35th ICRC*, (Busan)(2017).

[2] L. Allen, M. Rezazadeh, S. Meyer and A. V. Olinto for the JEM-EUSO Collaboration, *Proceedings of the 35th ICRC*, (Busan) (2017)

[3] W. Painter et al. for the JEM-EUSO Collaboration, *Proceedings of the 35th ICRC*, (Busan) (2017)

[4] L. Wiencke et al. fer the JEM-EUSO Collaboration, "The EUSO-SPB Mission and Science" *Proceedings of the 35th ICRC*, (Busan)(2017).

[5] J. Bayer et al. for the JEM-EUSO Collaboration, *Proceedings of the 33th ICRC*, #0432, (2013).

[6] V. Scotti and G. Osteria for the JEM-EUSO Collaboration, *Nuclear Instruments and Methods in Physics Research*, **A718** (2013), 248.

[7] G. Medina-Tanco et al. for the JEM-EUSO Collaboration, *Proceedings of the 32th ICRC*, #0301, (2011).