Available online at www.sciencedirect.com
SCIENCE DIRECT®

Nuclear Physics B (Proc. Suppl.) 150 (2006) 304–307

NUCLEAR PHYSICS B
PROCEEDINGS SUPPLEMENTS

www.elsevierphysics.com

Design, Implementation, and Performance of CREAM Data Acquisition Software

S.-Y. Zinn[a]*, H. S. Ahn[a], M. G. Bagliesi[b], J. J. Beatty[c], J. T. Childers[d], S. Coutu[c], M. A. DuVernois[d], O. Ganel[a], H. J. Kim[e], M. H. Lee[a], L. Lutz[a], A. Malinine[a], P. Maestro[b], P. S. Marrocchesi[b], I. H. Park[f], E. S. Seo[a], C. Song[d], S. Swordy[g], and J. Wu[a]

[a]Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742, USA

[b]Department of Physics, University of Siena and INFN, 53100 Siena, Italy

[c]Department of Physics, Pennsylvania State University, University Park, PA 16802, USA

[d]School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA

[e]Department of Physics, Kyungpook National University, Taegu, 702-701, South Korea

[f]Department of Physics, Ewha Womans University, Seoul, 120-750, South Korea

[g]Enrico Fermi Institute and Department of Physics, University of Chicago, Chicago, IL 60637, USA

Cosmic Ray Energetics and Mass (CREAM) is a balloon-borne experiment scheduled for launching from Antarctica in late 2004. Its aim is to measure the energy spectrum and composition of cosmic rays from proton to iron nuclei at ultra high energies from 1 to 1,000 TeV. Ultra long duration balloons are expected to fly about 100 days. One special feature of the CREAM data acquisition software (CDAQ) is the telemetric operation of the instrument using satellites. During a flight the science event and housekeeping data are sent from the instrument to a ground facility. Likewise, commands for controlling both the hardware and the software are uploaded from the ground facility. This requires a robust, reliable, and fast software system. CDAQ has been developed and tested during three beam tests at CERN in July, September, and November 2003. Recently the interfaces to the transition radiation detector (TRD) and to the timing-based charge detector (TCD) have been added. These new additions to CDAQ will be checked at a thermal/vacuum test of the instrument at NASA. The design, implementation, and performance of CDAQ are reported.

1. Introduction

CREAM is a balloon-borne experiment for measuring the energy spectra and composition of cosmic rays [1]. The energy of interest ranges from 1 to 1000 TeV and the elemental abundances will be measured from proton to iron nucleus. CREAM is scheduled to be launched from Antartica in December 2004. The CREAM payload contains, among other devices, a science flight computer (SFC) and two NASA flight computers called CDMs. Only one CDM is active at a given time and the other is a redundancy reserved for emergency. When triggered SFC gathers data from the detectors and passes them to the active CDM which will transmit the data by using either the TDRSS or Iridium satellite to an operations control center. The data will eventually be forwarded to a science ground computer (SGC) located at University of Maryland. Commands from SGC will travel the same route backward to SFC which will execute them.

CREAM consists of five detectors, namely, timing-based charge detector (TCD), transition radiation detector (TRD), silicon charge detector (SCD), hodoscope (HDS), and calorimeter (CAL). Event trigger and synchronization is or-

*Email: syzinn@umd.edu

0920-5632/$ – see front matter
doi:10.1016/j.nuclphysbps.2004.11.390

chestrated by the master trigger. TCD has nine concentrators which are running Linux operating system so SFC interfaces to TCD via TCP/IP. The interface to TRD is based on a FPGA board and a digital I/O board. CAL, HDS, and SCD are connected to SFC via custom PC/104 cards. The master trigger is connected to SFC by using another digital I/O board. In addition to the five detectors, SFC is collecting housekeeping data from a housekeeping board connected on a serial port. The housekeeping board monitors temperture, voltage, current, and pressure at totaling 386 channels. Also the housekeeping board reports the live and total time of the master trigger and the frequency counter output occurring from certain type of calibration runs. On a separate serial port, a command box is connected to SFC. The command box controls CAL, HDS, and SCD only. TCD and TRD commands are dealt by SFC through the interfaces described above.

2. Design

The CREAM data acquisition system (CDAQ) was designed with the following considerations. For its stability and multitasking nature, Linux was selected for the operating system of SFC. CDAQ should be simple, fast, and robust. For simplicity, the number of processes are kept at minimal level. Our goal for speed is to achieve 100 events per second. For long running, memory leak should be carefully checked. Whenever possible, we avoided using dynamic memory allocation not to introduce memory leaks at all. For easy maintenance and extensibility, programs are written in the C++ language. However, small portion of low-level routines was written in the C language. Graphical user interface (GUI) was also written for easy control of the detector. Finally, the data acquisition system should be usable not only for flight but also for laboratory and beam tests.

Figure 1 shows processes running on SFC and memory queues for interprocess communication. CDAQ_SERV is the master process that launches all other processes and handles signals to make sure that no zombie process appear. Threading is employed for CDAQ_SNIO to handle network communications effectively. One thread is dedicated to fragmenting science and housekeeping data. This is the PKT thread. The NW thread sends the packets resulting from the fragmentation of data by the PKT thread. The NW thread also receives packets and reassembles them. Commands from a ground computer are first received by CDAQ_SNIO and they are passed to CDAQ_CMD via the primary command queue. Some group of commands are executed by CDAQ_CMD. Commands related to housekeeping system are forwarded to CDAQ_HK which collects mainly the housekeeping data. All other commands are forwarded to CDAQ_EVT which interfaces to CAL, HDS, SCD, TCD, and TRD and builds science events. Science and housekeeping data from CDAQ_EVT and CDAQ_HK, respectively, are written to the data queue which are read by the packetizer thread of CDAQ_SNIO. The position of the CREAM instrument is tracked via the global positioning system (GPS). The NW thread of CDAQ_SNIO receives a GPS packet every second and stores it to the auxiliary data queue. Trigger rates are also written to the queue by CDAQ_EVT. The auxiliary data are read by CDAQ_HK and shipped out as a part of the housekeeping data.

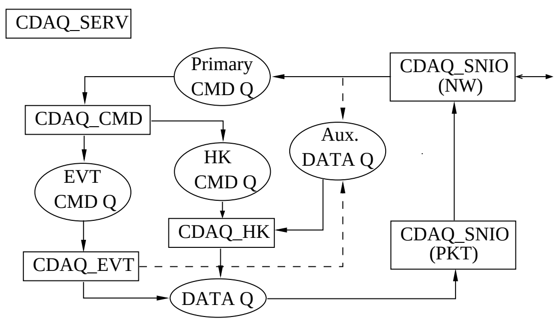

Figure 1. Server programs running on the science flight computer.

Figure 2 illustrates data flow on a ground computer and processes running on it. Data packets from SFC are received and acknowledged by CDAQ_CNIO. CDAQ_CNIO reassembles packets and extracts data. All data are saved first on a local disk. After that they are copied to three data queues on memory. For the purpose of online monitoring, one out of hundred events are copied to the data queue 1 associated with CREAM GUI program so that the GUI program displays an event about every second. But the reduction is not applied to queues 2 and 3. These two queues are for monitoring housekeeping data via HKMON and for relaying data to other computers for data backup and additional monitoring via CDAQ_RELAYD, respectively. Commands from the CREAM GUI are delivered to CDAQ_CNIO via a command queue. CDAQ_CNIO sends the commands to SFC.

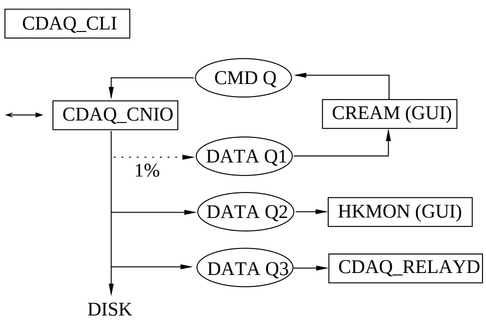

Figure 2. Client programs running on a science ground computer.

3. Implementation

3.1. Network communications

Now we describe the network communications between SFC and SGC. Among other network protocols, UDP is selected because it is connectionless so that it is easier to re-establish the communication in case of errors. However, custom protocol is necessary for maintaining connection and for delivering packets. Five types of fixed-length packets are adopted, namely, event data packet, housekeeping data packet, command packet, acknowledgment packet, and connection status packet. Often the length of science event and housekeeping data exceeds the packet size so fragmentation and reassembly are required. Each data packet contains four bytes of reassembly information. For maximal network bandwidth, packets are fully packed and event data can cross the packet boundary. Also a packet can contain multiple events.

When data packets are sent by SFC, it expects an acknowledgment packet from SGC within a second. SFC will send the same packet repeatedly until the packet is acknowledged or thirty attempts are made. After the thirty unsuccessful attempts, SFC drops the connection and checks SGC by pinging. SFC will resume sending the packet upon establishing a new connection. However, to prevent an infinite transmission loop, a packet will be timed out eventually and the next packet will be processed.

SFC sends the connection status packet once every five seconds to both CDMs and in return it receives the connection status packet only from the active CDM. By examining the packet, SFC determines which CDM is active and when a switch occurs it sends data to the newly activated CDM.

3.2. Interprocess communications

There are several methods for interprocess communication [2]. Message queue was first adopted for its simplicity. Queue size is 16 KB by default but can be enlarged. However, the length of data that can be written or read by a call is limited to 8 KB including four bytes of message header. Data exceeding the limit must be fragmented and reassembled. Data from beam or laboratory tests usually exceed this length limit. To ensure consecutiveness of fragmented data, a semaphore lock must accompany the message queue. Otherwise sophiscated algorithm for reassembly is required.

Message queue without a semaphore lock works faster than shared memory with a semaphore lock for data that are not exceeding the message size limit. However, a preliminary test shows that shared memory with a semaphore lock performs slightly better than message queue with semaphore. This may be due to the fact that with shared memory implementation fragmentation and reassembly are not required. Thus we adopted shared memory with semaphore whenever fragmentation and reassembly are required.

3.3. Prioritization of packets

Constant monitoring of the instrument status is vital. To ensure continuous flow of housekeeping data, packets are prioritized. Packets prepared by SFC for transmission are stored to different queues according to their priority. Housekeeping packets are of highest priority. The housekeeping packets convey housekeeping data, GPS data, trigger rates, live and dead times, data transmission rates, and error messages. Event packets come next in priority.

3.4. Event building

The master trigger is triggered by TCD, CAL, calibration board, or external device. The external trigger is employed for beam and laboratory tests. The event building process is branched according to the trigger. CAL, HDS, and SCD data are read always because the four triggers are valid for them. TCD and TRD data are read when the event is triggered by TCD.

Sparsification of data is performed at hardware level by all detectors. Additional sparsification is done by the data acqusition process for CAL, HDS, and SCD for a special type of events.

Data from the detectors contain unique event number distributed by the master trigger. The event building process compares the event numbers for checking the synchronization of the detectors. Upon finding three consecutive mismatches, it resets the instrument.

3.5. Operation modes

To use the same data acquisition system for flight, beam and laboratory tests, three modes of operation are provided. Three operation modes are based on the same event building routine. Thus laboratory or beam test should be sufficient for checking the integrity and performance of the data acquisition system. All commands are executed in the same manner regardless of the mode.

In flight mode, data are taken continuously and as fast as they can be. At preset intervals, different type of calibrations are performed automatically. This serves primarily checking pedestal drift of data channels and provides charge calibration data. In test mode designed for beam and laboratory tests, data acquisition occurs every second. In test mode the data acquisition software acts like an oscilloscope and one can check the various components of the instrument easily by examining the on-line display. Data can be taken for given number of events or continuously in test mode also. But calibration runs are not kicked in automatically. Finally in standby mode CDAQ waits for commands. No autonomocity is associated with standby mode.

4. Performance

CDAQ has been tested daily at laboratory and three times in 2003 during beam tests at CERN. During the beam tests, it performed well without a single failure. Full readout of CAL, HDS, and SCD without sparsification was done at peak speed of 125 events per second: this corresponds to 3 MB/s. No software crash was experienced during the beam tests.

5. Summary

CREAM data acquisition has been implemented primarily using the C++ language for modularity and extensibility. CDAQ has been tested daily in laboratory, which provides a good testbed. Beam-test operations show the stability of CDAQ. Also the acqusition speed is satisfactory. However, with recent additional interfaces with TCD and TRD, the speed is expected to drop accordingly but not significantly. Full integration and operation of the detectors will be done shortly and reported elsewhere.

REFERENCES

1. E.-S. Seo et al., Advances in Space Research **30**, 1263 (2002).
2. W. R. Stevens, UNIX Network Programming (Prentice Hall, New Jersey, USA, 1990), chap. 3.