33RD INTERNATIONAL COSMIC RAY CONFERENCE, RIO DE JANEIRO 2013
THE ASTROPARTICLE PHYSICS CONFERENCE

ICRC 2013

# The Data Acquisition System for CREAM on the International Space Station

D. ANGELASZEK[1,2], Y. AMARE[2], C. EBONGUE[1,2], R. GUERCI[1,2], M. GUPTA[2], H.G. HUH[2], M.H. KIM[2], M.H. LEE[2], L. LUTZ[2], E.S. SEO[1,2], Y.S. YOON[2]

[1] *Department of Physics, University of Maryland, College Park, MD 20742 USA*
[2] *Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742 USA*

*dja1987@umd.edu*

**Abstract:** The balloon-borne Cosmic Ray Energetics and Mass (CREAM) Experiment has carried out six successful flights around the continent of Antarctica for a total exposure of over 160 days. The CREAM Data Acquisition (CDAQ) software system, utilized on all six balloon misions, is a crucial component of the CREAM instrument that facilitates data-taking, monitoring, commanding and calibration of the entire apparatus. Currently, a CREAM payload is being developed for integration on the International Space Station (ISS) in 2014. Numerous CDAQ modifications were required to accommodate updates to the instrument for event gathering, data managing, networking and commanding processes and to meet ISS requirements. Additions were also made to incorporate new instrument detectors. This latest version of CDAQ, including its relation to the original design and balloon flight implementation, is discussed in detail.

**Keywords:** CREAM, data acquisition, ISS

## 1 Introduction

The Cosmic Ray Energetics and Mass (CREAM) experiment carried out a series of six balloon flights over the continent of Antarctica from 2004 to 2010 [1]. The CREAM instrument consisted of a science flight computer (SFC) and a suite of particle detectors designed to measure the charge and/or energy of cosmic ray nuclei during these flights. A system of custom designed common electronics were also incorporated into the instrument to ensure that the various payload components performed cohesively [2]. The SFC collected science and housekeeping data from the detectors during flight. The science data consist of both physics and calibration events that measure the in flight response of the detectors. Housekeeping data is used to monitor the overall health and status of the instrument and its components. The SFC also executes commands and requested calibrations issued from ground operators. Its command and data taking responsibilities are facilitated by the CREAM Data Acquisition (CDAQ) software system [3]. Similar to the electronics system, CDAQ is custom designed to meet the requirements of the individual payload.

A follow-on project has been underway since 2010 to integrate a CREAM detector on the International Space Station (ISS). This project is known as Cosmic Ray Energetics and Mass on the International Space Station or ISS-CREAM [4]. The radiation environment of the ISS is significantly more hostile than conditions experienced on the balloon flights, so the electronics of the instrument have been upgraded to account for the increase in ambient radiation [5]. In addition, the command & data handling (C&DH) environment of the ISS is significantly different from the NASA support systems and interfaces used for balloon flights. Lastly, ISS-CREAM has been designed to last for at least three years of continuous data taking, and there are currently no plans for payload recovery at the conclusion of its mission. These operational considerations are significantly different from the balloon payloads that were recovered in Antarctica after a 6-42 day flight. Several upgrades were made to the CDAQ software system in order to accommodate the transition from balloon flights to the ISS. The hardware, operational and environmental changes that necessitated these updates, as well as testing being done to verify their functionality, will be discussed.

## 2 Software Design Implementation

CDAQ is a modularized system written in C++ that runs on a Linux-based operating system. It is based on a client-server model [6]. The server programs run on board the SFC that is patched with Real Time Application Interface (RTAI) software that helps reduce run-time latencies. The CDAQ client runs on a ground machine located in the Science Operation Center (SOC) at the University of Maryland (UMD). CDAQ accomplishes its numerous tasks by utilizing a multi-process, multi-threaded architecture, which is shown in Figure 1.

On the CDAQ server, CDAQ_SERV is the master process that generates and maintains the other processes. CDAQ_EVT gathers science data from the detectors and builds events. CDAQ_HK gathers housekeeping data from the housekeeping system. CDAQ_CMD interprets and executes commands received from the ground. CDAQ_SISSNIO_PLAYBACK handles the responsibilities of a custom designed playback system. This playback system is discussed in detail in section 3.3. CDAQ_SISSNIO handles all network related responsibilities, such as packetizing data, sending the data to the ground and receiving commands issued from the SOC. It also saves science event packets to disk. The various processes communicate with each other through shared memory character queues, represented as ovals in Figure 1.

CDAQ_SISSNIO and CDAQ_SISSNIO_PLAYBACK have tasks that need to run in parallel and independent of other tasks. Therefore, the two processes divide up their tasks into threads which can run concurrently. On CDAQ_SISSNIO, the PKT0 and PKT1 threads take housekeeping and science event data, respectively, and generate the network packets that will be sent to the ground. The

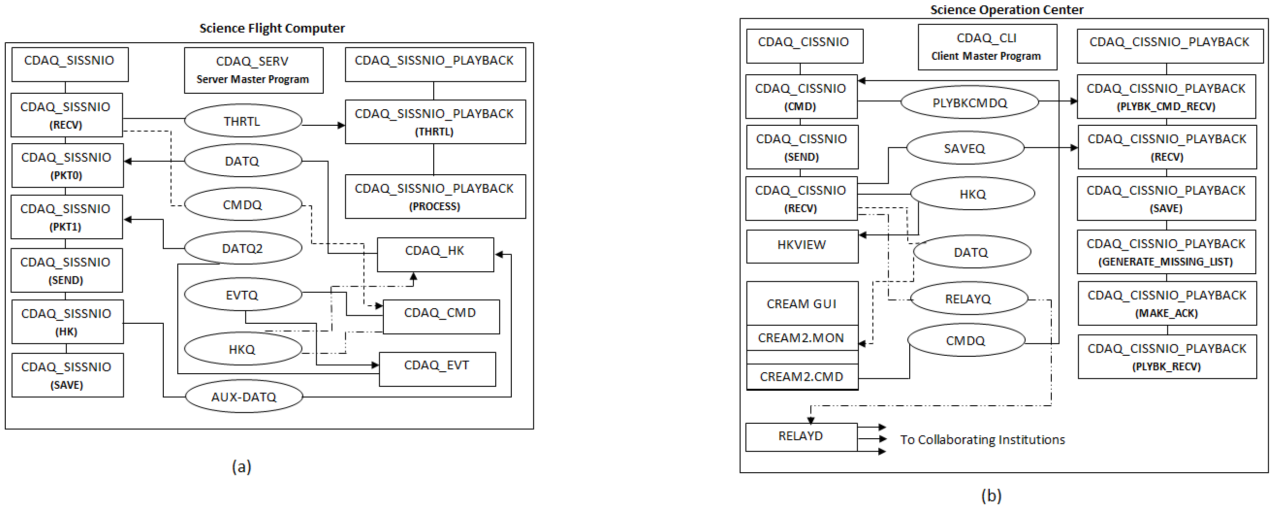


**Fig. 1**: Schematic representation of the various connections between the server's (a) and client's (b) processes. Dotted lines are used for distinguishing different paths that cross each other and individual threads are shown connected to their parent process.

HK thread relays information on the capacity of memory buffers used to store housekeeping, science and packet data to CDAQ_HK. The SEND thread is responsible for sending network packets, while RECV is responsible for receiving commands sent from the ground and relaying them to CDAQ_CMD or CDAQ_SISSNIO_PLAYBACK. SAVE is the thread that stores science data network packets to disk. CDAQ_SISSNIO_PLAYBACK utilizes two threads. PROCESS receives playback requests for missing packets not received on the ground, processes the request and transmits the appropriate packets. THRTL receives throttling information from CDAQ_SISSNIO. Throttling is discussed in more detail in section 3.2.

For the CDAQ client, there are two main processes (CDAQ_CISSNIO and CDAQ_CISSNIO_PLAYBACK) and 4 independent programs (CREAM2.MON, CREAM2.CMD, HKVIEW and RELAYD). The CDAQ Graphical User Interface (GUI) is comprised of the programs CREAM2.MON and CREAM2.CMD. CDAQ_CISSNIO has three threads. CMD receives requests to send commands from CREAM2.CMD and constructs them. RECV receives all data sent from the server and copies packets containing science event data to CDAQ_CISSINIO_PLAYBACK. RECV will send about 1% of all science data to the GUI's CREAM2.MON and all housekeeping data to HKVIEW, so that SOC operators can monitor real-time data during flight operations. RELAYD also receives science data from RECV and stores it so it can be made available to collaborating institutions. SEND transmits commands to the server. On CDAQ_CISSNIO_PLAYBACK, the RECV, SAVE, GENERATE_MISSING_LIST and MAKE_ACK threads receive science event packets from CDAQ_CISSNIO, save them to disk, determine data packets that are missing and generate the request for missing packets, respectively. The PLYBCK_CMD_RECV thread receives signals from the CDAQ_CISSNIO's CMD thread that SOC operators have requested to begin a playback session. Playback sessions are discussed in section 3.3.

# 3 CDAQ Updates for ISS Integration

## 3.1 Hardware Design

The CDAQ software system is custom designed to meet the requirements of the CREAM instrument, so changes to instrument hardware require updates to CDAQ as well [7]. A schematic overview of the CREAM hardware layout is shown in Figure 2. One noticeable feature of the figure is the presence of two SFCs to supply redundancy in case one computer fails. The top of the diagram shows the individual electronic boards responsible for taking data from the various detectors: the calorimeter (CAL), silicon charge detector (SCD), which actually consists of four separate boards for each of the four layers of sensors for the detector, top and bottom counting detector (TCD/BCD) and boronated scintillator detector (BSD). The TCD/BCD, BSD and 2 layers of SCD sensors were added for ISS-CREAM. They were never flown on the balloon flights.

Electronic boards for the master trigger system (MT), housekeeping system (HK) and command system (CMD) are also shown. The Instrument Interface Unit (IIU) is a new electronic board designed to monitor the SFCs and power them on and off. The IIU board will also be able to select the boot source for each SFC. Each SFC will have 4 boot sources for redundancy. All of these boards are connected to the SFCs via two independent USB hubs that are part of another electronics board called the SFC Interconnect Box. As part of the new electronics design for ISS-CREAM, the command board is now connected via USB to the SFCs. The command board and the SFC's were connected via an RS-232 Serial connection on the balloon flights. The switch to a USB based connection required CDAQ's USB interface to be extended to the software command process.

The hostile radiation environment of the ISS will occasionally result in single event upsets (SEUs) that disrupt the individual electronic boards. During the balloon flights, SEUs would cause a CDAQ system crash even if the problem was localized to an individual board. Given the increased frequency of these events on the ISS, CDAQ was updated to prevent a crash due to a single board experiencing an SEU. Furthermore, the SFC Interconnect Box is being modified to allow CDAQ to individually reset inactive

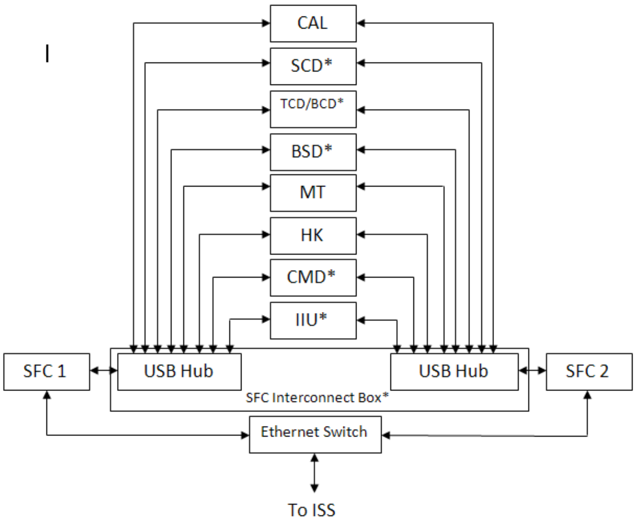


**Fig. 2**: Schematic of the hardware design for ISS-CREAM. Asterisks are used to show components that are new or updated for ISS-CREAM.

boards. After a reboot, CDAQ will then re-establish the USB connection between the SFC and the board.

### 3.2 ISS Command & Data Handling Environment

For the balloon flights, NASA provided its own independent C&DH support interface. The interface was known as the Command Data Module (CDM) for CREAM I-IV and the Support Instrumentation Package (SIP) for CREAM V-VI. The main difference between CDM and SIP is that the CDM communicated with the SFC's via Ethernet-based protocols, whereas SIP communicated with the SFC using a Serial-based connection. The main advantage provided by the NASA support interface was that it served as the interface between the CREAM instrument and the NASA systems needed to send information to ground. Consequently, CDAQ's network processes could be written using standard and well-known Ethernet-based (such as the User Datagram Protocol (UDP)) or Serial-based protocols, while remaining ignorant of the NASA-specific protocols employed for communication to the ground. There will be no NASA C&DH support interface provided on the ISS, so CREAM must directly interface with the ISS network.

Commands sent from ground-to-payload and telemetry sent from payload-to-ground would have to utilize two different networks on board the ISS. The commanding or uplink path uses the MIL-STD-1553B (or 1553 for short) protocol, whereas the telemetry or downlink path is Ethernet-based. Both of these networks use the Consultative Committee for Space Data Systems (CCSDS) protocol. The use of two separate datalinks is required because the Ethernet-based downlink path using the CCSDS protocol is unidirectional and cannot be used to uplink commands. The 1553 based datalink is bidirectional, but the downlink data rate was determined to be too small for ISS-CREAM. Since CDAQ was written for UDP based Ethernet and Serial based communications, the differences inherent to the ISS C&DH environment would require a significant revision to the software system.

A significant rewrite to CDAQ was undesirable given its successful performance using UDP and Serial based communications on the six balloon flights. The significant rewrite was avoided through use of the Software Toolkit for Ethernet Lab-Like Architecture (STELLA) developed by The Boeing Company [8]. STELLA is a software application designed to serve as a UDP or TCP (Transfer Control Protocol) interface between a payload's software and the ISS C&DH systems. For ISS-CREAM, STELLA will be installed directly on the SFCs. CDAQ will then directly interface with the STELLA application for sending data and receiving commands using UDP and can remain ignorant of the 1553/CCSDS protocols and formats required for the ISS. STELLA will be installed on a dedicated interface machine running a NASA application known as the Telescience Resource Kit (TReK) at the SOC [9]. The purpose of TReK is to serve as a secure interface between a payload's operation center and the Huntsville Operation Support Center (HOSC) located at the Marshall Space Flight Center (MSFC). The HOSC is the NASA ground station that receives payload telemetry from the ISS and routes data to the TReK machines at the appropriate payload operations center. Telemetry received by TReK at the SOC will then be forwarded to STELLA, where it will be processed and sent to the CDAQ client machine in the expected packet format generated by the CDAQ server. A similar process works in reverse for sending commands to the CREAM payload. As a result, STELLA will provide a seemingly transparent Ethernet-based connection between server and client that will allow CDAQ to retain its proven UDP based networking capabilities, as show in Figure 3.

Unlike the balloon flights, ISS-CREAM will be one of many payloads using the resources of the ISS. One of these resources is the available downlink bandwidth which is shared among all payloads using the same datalink. Therefore, there may be times when ISS-CREAM is required to temporarily lower its data rate due to another payload's request for more bandwidth. In order to accommodate this, CDAQ was given the ability to adjust, or throttle, its data output rate by sending a command from the ground that adjusts the rate at which the CDAQ server transmits packets. Throttling information is also relayed to the CDAQ custom playback system to ensure the playback data rate is adjusted accordingly.

### 3.3 Operation Requirements for ISS-CREAM

Occasional packet loss is expected during the transmission from server to client. On the balloon flights, the NASA C&DH interface was directly connected to the SFC, and it could store all CREAM data packets. The missing data packets could be retrieved when the payload was recovered. For ISS-CREAM, there will be no NASA C&DH interface to store data packets, and there is no plan to recover the CREAM payload. In order to maximize the amount of science data received from the payload, ISS-CREAM will rely on playback systems to retrieve lost data packets. There will be four separate methods for retrieving packets.

Two of the playback methods are services provided by the High rate Communications Outage Recorder (HCOR) and the HOSC. HCOR will playback data that was sent from payloads during communication outage periods between the ISS and the ground [10]. The HOSC will store all data received from ISS payloads and make it available for playback using HOSC web applications. These two systems are both external to CREAM, which limits their playback capability. This is because data packets dropped before

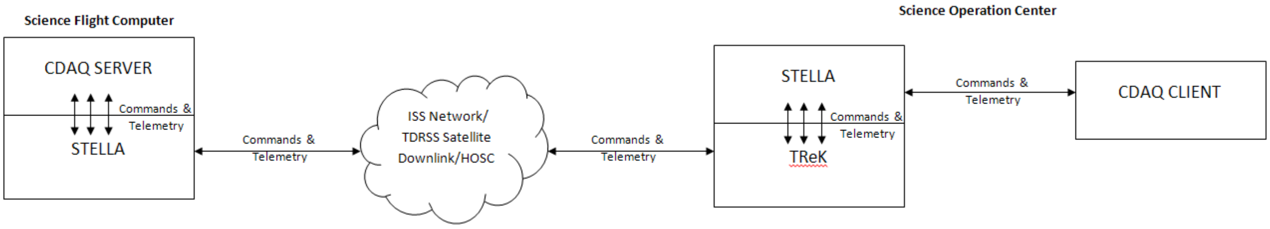


**Fig. 3**: STELLA provides the illusion that the CDAQ server and client are directly connected via a UDP-based Ethernet connection. This allows the complicated network connections between the ISS, Tracking and Data Relay Satellite System (TDRSS) and the HOSC to be abstracted away from CDAQ.

reaching the HOSC or HCOR will not be available for playback via these systems.

In order to account for this, science event packets are stored directly on the SFC, which will possess three 32GB flash memory devices for storing data. Flash memory has the limitation that it allows a finite number of write/erase cycles (usually in the range of $10^5$ to $10^6$ cycles) before the device fails. During flight, CDAQ is expected to generate 33 science event packets per second. At this rate, CDAQ would quickly use up and cause the failure of the flash memory devices. To account for this, the CDAQ server first stores individual science data packets to the Random Access Memory (RAM) of the SFC. After $2^{15}$ packets have been stored to RAM, the packets are written to the flash memory device currently in use as a single file called a packet group file. This procedure will ensure that $10^5$ writes to a single flash memory device will take more than 3 years.

CDAQ packets contain a 32-bit packet serial number that increments each time a packet is transmitted. The CDAQ client can keep track of the packets that were not received from the server by identifying breaks in the packet serial number. After a period of time corresponding to when a packet group file is generated, the CDAQ client generates a list of missing packets. SOC operators can then request a playback session using the CREAM GUI, and the client will begin sending requests to the server to resend the missing packets. By referring to the packet serial number requested, the server can quickly access the appropriate packet group file, retrieve the desired packet, and re-transmit it to the server. As playback packets are received, the client removes packets from the missing packet list.

In addition to this custom designed system, STELLA offers a file transfer service that can also be used for playback. The SOC operators will be able to copy an entire packet group file from the SFC to the ground with this service. The group file can then be sent to the CDAQ client for processing. The STELLA file transfer service and the custom designed CDAQ system offer the advantage that there will always be a way to recover packets lost during the transmission process unlike the HOSC and HCOR.

In order to prevent the flash memory devices from filling to capacity, a method of deleting the packet group file must be made available. This can be done using a specialized command from the CDAQ client or by using another STELLA service known as the STELLA console service, which allows SOC operators to access the server through a Linux terminal-like interface.

## 4 Testing and Further Development

Development testing of the CDAQ software system was conducted at the Space Systems Integration and Test Facility (SSITF) at MSFC in the spring of 2013. The testing was designed to simulate the actual C&DH setup ISS-CREAM will use on the space station by connecting an SFC computer to the Payload Rack Checkout Unit. A connection between the SFC computer at SSITF and the CDAQ client machine at SOC that allowed for a steady, reliable flow of commands and telemetry was established during the testing. These results help confirm that the planned CDAQ software system is compatible with the C&DH systems of the ISS. A finalized ISS-CREAM CDAQ is planned for completion by late summer of 2013.

**Acknowledgment:**This work was supported by NASA Headquarters under the NASA Earth and Space Science Fellowship Program-Grant NNX12AL57H, NASA grant NNX11AC52G and their predecessors.

## References

[1] `http://cosmicray.umd.edu/cream/`.
[2] Ahn, H. S. et al., The Cosmic Ray Energetics and Mass (CREAM) Instrument, Nucl. Instrum. Methods, A579, 1034-1053, 2007.
[3] Zinn, S. Y. et al., The data acquisition software system of the 2004/2005 CREAM experiment, Proc. of the 29th International Cosmic Ray Conference, Pune, India, 3, 437-440, 2005.
[4] Seo, E. S. et al., ID 629 (2013) this conference.
[5] Amare, Y. et al., ID 630 (2013) this conference.
[6] Zinn, S. Y. et al., Design, Implementation and Performance of CREAM Data Acquisition Software, Nucl. Phys. B (Proc. Suppl.), 150, 304-307, 2006.
[7] Yoo, J.H. et al., Improved Data Acquisition System for CREAM-III, Proc. of the 30th International Cosmic Ray Conference, Merida, Mexico, 2, 401-404, 2007.
[8] International Space Station-ExPRESS Payload Simulator. 5 December 2012. NASA. `http://www.nasa.gov/mission_pages/station/research/experiments/ExPRESS_Payload_Simulator.html`. 22 May 2013.
[9] Telescience Resource Kit. 13 November 2012. `http://trek.msfc.nasa.gov/`. 22 May 2013.
[10] Mixson, C. D., Operations Methodology for the International Space Station (ISS) High Rate Communications Outage Recorder (HCOR). `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19990100661_1999163697.pdf`. 22 May 2013.