Web-Based Monitoring of BARREL Flights

Warren Z. Rexroad[1] and David M. Smith[2]
Santa Cruz Institute for Particle Physics, University of California, Santa Cruz, CA 95064

Robyn M. Millan[3]
Department of Physics and Astronomy, Dartmouth College, Hanover, NH, 03755

The Balloon Array for RBSP Relativistic Electron Losses (BARREL) is part of NASA's Living With a Star program. In order to support the upcoming BARREL flights, a system for monitoring incoming telemetry in real time is a necessity. The work on BARREL is being done at universities distributed across the US by a number of people who would like to have access to the incoming data. The BARREL Science Operations Center (SOC) was created to provide near real-time, remote data access to the widely-distributed science community. From a web-based interface any user is able to view telemetry data for all balloons in the array, generate strip charts with a variety of plotting options, produce dynamic flight maps, monitor payload health and alerts, and download ASCII data files by date. All tools are written in HTML5/JavaScript so no special software is required. This allows everything to be accessed using a computer or device with a modern browser. This paper will discuss the the capabilities of the BARREL SOC and provide information about why certain decisions were made in hopes of aiding others designing a similar system.

I. Introduction

In an effort to study the dynamics of the outer radiation belt, NASA will be launching RBSP (Radiation Belt Storm Probes) as part of their Living With a Star Program. The instruments on board RBSP will measure the particles in the belt along with electric and magnetic fields. However, the space probes will not be able to determine which electrons are destined to be lost to Earth's atmosphere. In order to properly characterize loss events, a low-altitude detector must be flown. BARREL (Balloon Array for RBSP Relativistic Electron Losses) will fill this need.

A balloon array was chosen as a low altitude detector because it provides a number of benefits including lower cost, having a relatively static position, and the certainty of only detecting electrons that are lost to the atmosphere. In addition, the main advantage provided by a balloon array, rather than a low orbiting satellite, is the ability to take multi-point measurements of relativistic electron precipitation. These types of measurements will allow the study of both the spatial and temporal evolution of precipitation events in a way that has not been done before.

In each BARREL campaign, taking place during the Austral summer seasons of 2012 and 2013, 20 payloads will be flown around Antarctica. These payloads will be divided between the South African Antarctic Station (SANAE IV), located at 2.8°W, 71.67°S, and the British Antarctic Survey’s Halley Bay station, located at 26.5°W, 75.5°S. From the launch site, they will float to approximately 35km and be carried westward around the pole at rate of 20 to 40 knots by the circumpolar winds. With a scheduled launch every 1 to2 days, and an expected flight time of 5 to 10 days per balloon, an array of 4 to 5 detectors will be spread of 150° of geographic longitude. Magnetic field lines connect the balloons with the heart of the radiation belt at Earth's equator. Measurements made by BARREL in conjunction with the in situ measurements from RBSP provide valuable information about two important electron loss events: Duskside Relativistic Electron Precipitation (DREP), and Relativistic Microbursts. DREP events have an observed spatial distribution of ~15° of geographic longitude and microbursts have been observed over ~90° geographic longitude. Something that is still unknown about these events is how their spatial distribution evolves in time. The detector array of by BARREL will provide a unique opportunity to simultaneously observe these events from several locations.

During loss events, energetic electrons entering the atmosphere are deflected and emit X-rays (known as bremsstrahlung, or braking radiation). Each BARREL payload will carry instruments to measure X-rays and

[1] Science Operations Center Software Manager, SCIPP/UCSC, 1156 High St., Santa Cruz, CA, 95064, USA
[2] Ground Station Manager, SCIPP/UCSC, 1156 High St., Santa Cruz, CA, 95064, USA
[3] Principal Investigator, Department of Physics and Astronomy, Dartmouth College, Hanover, NH 03755 USA

magnetic field fluctuations during these events. Bremsstrahlung will be measured using a single 3” x 3” NaI scintillator that provides data at three time resolutions. The fastest time resolution measurements will be binned into four energy channels, the medium will create a 48 channel spectrum, and the slowest will provide a 256 channel spectrum. A Bartington magnetometer will be on board for identifying substorm activity and ULF oscillations of the magnetic field. Position and time will be tracked with a Lassen-SQ GPS. An ADC will also be included to convert analog housekeeping data including voltage, temperature and current consumption of all subsystems. All data will be will formated into 214 byte frames and transmitted to the BARREL ground station via Iridium modem.[1]

All of these data will need to be monitored continuously to watch for any interesting events, equipment failures, or the payload drifting out of the approved flight area. The BARREL ground station will collect, process, and disseminate all data in near real time through a publicly available web interface. In this paper we present an overview of the BARREL ground station operations. Section 2 provides a general description of how the ground station is configured and gives a reasoning for the basic layout. Section 3 lists the requirements for a multiple payload ground station and describes how they are filled. Section 4 gives an in depth look at the ground station components that process the data frames and provide web tools to the science team and public.

II. BARREL Ground Station

The BARREL ground station is a distributed network of computers separated into two main groups: Mission Operations Center (MOC) computers and Science Operations Center (SOC) computers as in Fig. 1. The BARREL MOC is used for contacting the payloads via Iridium modems to collect data or (if deemed necessary) issue cut down commands. Each MOC computer will handle up to four payloads at a time and will write data frames retrieved from the payloads to binary files on both a network-attached storage device (NAS) and its internal drive for redundancy. Multiple computers store their data on the NAS without conflict by giving each payload a dedicated directory. Inside each payload directory data files are further sorted into date directories. For security reasons, both physical and network access to the MOC are highly restricted. The room that houses the MOC hardware is always locked and the key is held by no more than three lead team members a time. The MOC computers run Windows XP with all network services disabled with the exception of the ability to write to the MOC NAS using SMB/CIFS protocol. Furthermore, all MOC components are connected to the university network behind a hardware firewall.

The BARREL SOC reads data frames from the MOC NAS shortly after they are written and outputs multiple ASCII files (CSV format) to the SOC NAS. Each ASCII file contains a single data type (e.g. magnetometer readings, X-ray spectra, GPS coordinates, etc.) and is an accumulation of the data in MOC's binary files gathered over the course of a day. These data files are what will be viewed by the public and science team when using the SOC's web-based tools.

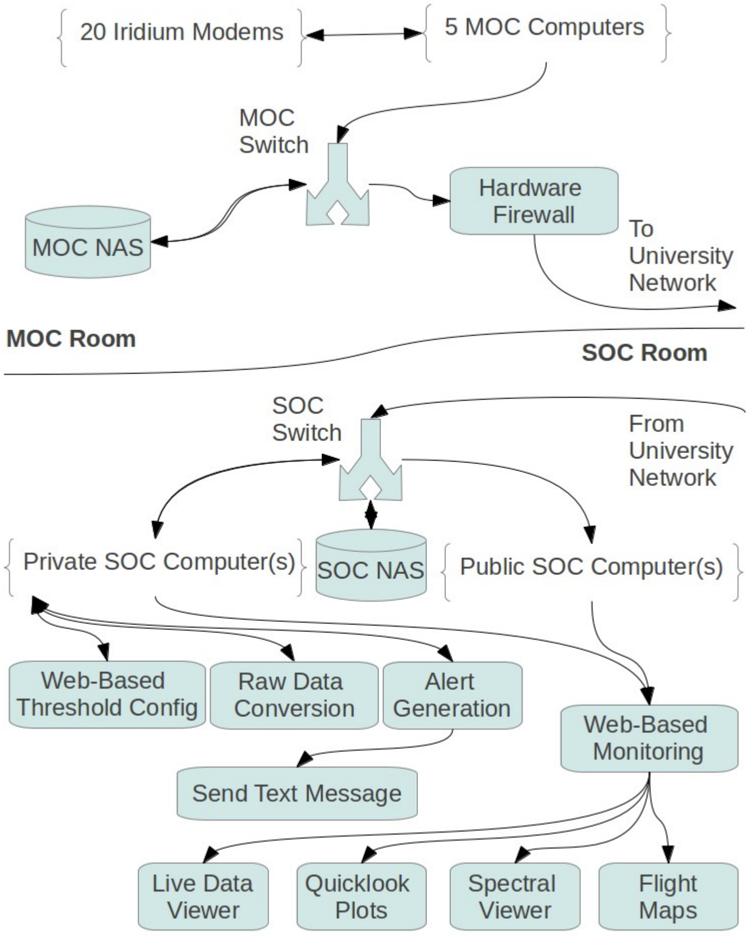


Figure 1. BARREL Ground Station Hardware. *Network connections of the MOC and SOC are shown along with the SOC's available functions in the private and public configuration.*

As with the MOC, all files written by the SOC are stored on a local NAS where payload-specific files are split into dedicated payload directories. In addition to the ASCII data files, the SOC also writes a number of small payload status files that contain information such as the current working date and the last binary file read. These status files allow coordination between multiple SOC computers and the various SOC programs that are running.

Unlike the MOC, the SOC also stores a number of common configuration files that are used to control the multiple software components. These files also use the SOC NAS as a central storage location, but are in a common directory, rather than in payload-specific directories. Figure 2 shows how all of the SOC and MOC files are stored on the network.

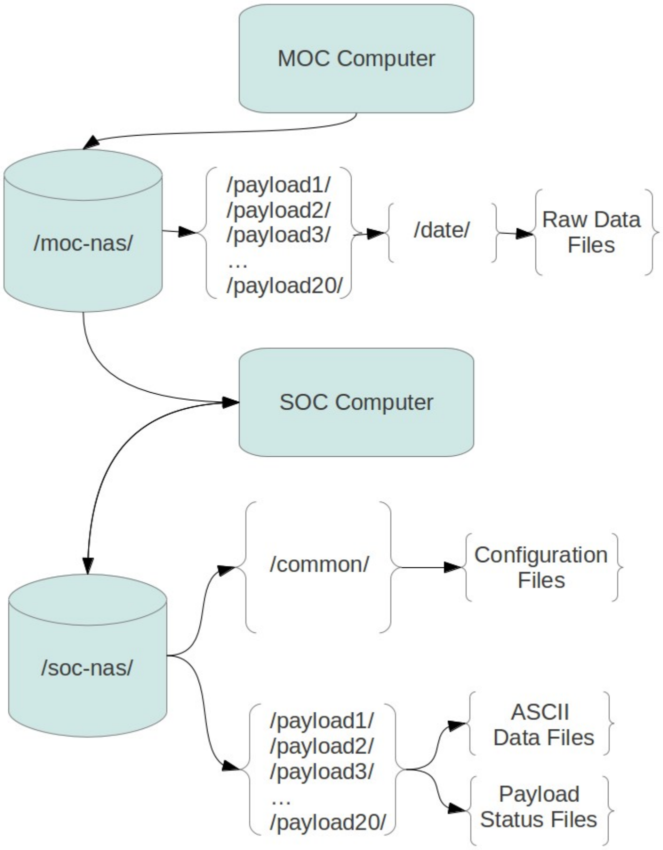


Figure 2. BARREL Ground Station File System. All of the files on the MOC and SOC are shared with each other through networked-attached storage devices.

III. Feature Overview

In order to support up to 20 payloads simultaneously, there are a number of features that must be included in a telemetry monitoring package. This section will outline those features most important to provide continuous monitoring of a multiple payload mission.

1. Remote Monitoring

Modern web technologies make it possible to have a fully functioning telemetry monitoring system contained wholly in the browser (requiring no special software or browser plug-ins to be installed). This eliminates the need for a person to physically sit behind an on-site terminal 24 hours/day to monitor payload activities. Furthermore, the full team and the interested public will be able to see all data returned by the payloads in near real time. BARREL, having participating universities located all across America, will greatly benefit from this method of telemetry monitoring. The task of staffing Mission Monitors is also greatly eased by expanding the pool of potential workers to anyone with a computer and reliable internet connection, rather than just those local to the ground station (specifically Santa Cruz, California).

For security reasons, BARREL will restrict MOC access to just a few team members. The only reasons they will access the MOC are to perform maintenance and payload cut-down procedures. All telemetry monitoring will be done through the SOC via web interface. The SOC provides a number of web-based data visualization tools that will be available to both the Mission Monitors and the public.

2. Simultaneous Data Processing for Multiple Payloads

The ability to process data that are being sent from multiple payloads is of the utmost importance for successful telemetry monitoring. This is achieved in the BARREL SOC by having the main data processing program (updater.pl) run as an independent process for each payload. Each process reads raw data files from a payload-specific directory on the MOC NAS and writes processed output files to a payload-specific directory in the SOC NAS. The reason for using network-attached storage devices for the input and output files is twofold: first, having the MOC write its raw data files to a NAS fulfills a security requirement for the MOC computer to be isolated behind a firewall with no direct communication with another computer; second, both input and output files stored on a NAS allow for multiple SOC and MOC machines to be running and sharing a common directory tree.

The MOC is planned to have no more than four payloads per machine (up to five machines will be running to support 20 payloads), however, there is no such limitation on the SOC at this time. The SOC machine was chosen to have two quad-core processors, meaning that under full load each processor core will handle no more than three instances of the program. Initial testing has shown that this load will not over-stress the machine's resources. If, however, a single computer is found to be overly burdened, the common file system on the NAS makes adding SOC computers a trivial task.

3. Security and Reliability

Even though it is convenient to provide all of the telemetry monitoring tools on the web, this does pose some reliability risks. First, if only one computer is being used for all SOC operations, web server traffic may bog down the processing of raw data files. This is avoided by having multiple SOC machines, one private and one public. The publicly available SOC, used by off-duty Mission Monitors and the public, will only serve a website and read data from the SOC NAS. On-duty Mission Monitors will use a password protected website served by the private SOC machine(s). These private machines will also be responsible for processing raw data and writing ASCII files to the SOC NAS. The separation between public and private machines will ensure the Mission Monitors have reliable access to the data without any worry of slowing down critical processes with too many web requests.

The SOC has been designed and tested in a Linux environment and will operate behind a firewall. All of the server-side programs have been custom written making an effort not to use standard libraries that may have well-known vulnerabilities. This will increase security by avoiding attacks generated by random IP/port scanners that attempt to exploit poorly implemented services. In the unlikely event of a successful malicious attack, the SOC can easily have external access to its webserver turned off and have all of the telemetry monitoring performed locally.

Another reliability concern that may arise is a power outage. Laptops were chosen for the MOC computers so they could continue to run off of their internal batteries in the event of a power failure. The Iridium modems and MOC NAS will be powered with an UPS to ensure continuous operation. The SOC computers, on the other hand, will not have any backup power supplies. The decision to omit backup power for the SOC was made because it would provide the science team with the most effective form failure notification. That is, should the SOC suddenly become unresponsive the Mission Monitors will be altered to a problem that must be dealt with immediately. With all telemetry monitoring being conducted through a web interface, there may be times when no one is physically on site to see there has been a power outage. However if the SOC is suddenly unresponsive (either due to a network, hardware, or power failure) the Mission Monitor will immediately notify the duty scientist and the situation can be assessed. This method of failure detection will allow for ample time to prepare for long term outages while the MOC hardware will continue to operate and record data. The SOC has the ability to processes data at a rate faster than it is transmitted to the MOC, so any files that are written during a SOC outage can still be brought current once the system is restored.

4. Payload Health Monitoring and Alert Generation

Even though it is planned to have no less than two Mission Monitors on duty at any time, it is still an arduous task to monitor up to 20 payloads each returning over 50 data products continuously. Each payload must be monitored to ensure it is not flying out of range, power is not failing, and to watch for interesting data. To aid the Mission Monitors in their duty, an alert generation system is included in the BARREL SOC. The alert generator will monitor the end of output files stored on the SOC NAS and check for out-of-range data. There are two types of alerts that may be generated, yellow alerts and red alerts. The yellow alerts are simply to draw the Mission Monitor's attention to interesting data, or possibly a failing sensor (i.e. something that should be recorded in their report, but does not require immediate action). Red alerts indicate a more serious problem, such as a power failure, a balloon flying out of approved airspace, sinking too quickly, low altitude, or a payload that has stopped returning data for a significant amount of time. These red alerts generally require action from the duty scientist and cannot be ignored. In order to make sure there are no “asleep at the wheel” situations with Mission Monitors, upon finding a red alert, the generator will display a message on screen. The Mission Monitor must clear the message withing a given time frame, or the SOC will contact the duty scientist directly via text message and/or email.

To help the Mission Monitor quickly identify yellow and red alerts, a color coded status bar is provided that gives details on each payload via tool tip as in Fig. 3.

5. Data Visualization

In order for a telemetry monitoring system to be useful, it must give a number of different ways to view incoming data. The BARREL SOC will provide the following tools for working with the near-real time data:

1) Live Data Viewer – Displays the most recent of each data product returned by the payload.
2) Quicklook Plots – The most recent one, six, or twenty-four hour period of data for a particular product displayed in a strip chart.
3) Spectral Viewer – Histogram view of the two X-ray spectra returned by the payloads integrated over a specified period of time.
4) Maps/Payload Tracker – A map of Antarctica with an overlay of each payload's path during a specified time period.

Each of these tools will be discussed with more detail in the following section.

IV. SOC Components

There are two main groups of SOC components: data processing and data visualization. Generally, each component is a separate program and can be run as multiple instances at a time, sometimes from multiple computers. The use of the SOC and MOC NAS devices allow all of these independently operating programs to coordinate with each other by sharing the same set of data files and being able to read/update a set of common status files. This section will describe how each of these components function and use the common file systems.

1. Binary File Processing (Updater)

All of the BARREL SOC's functionality starts at its ability to collect and process the raw data from the MOC in near real time. Each call the MOC makes to a payload results in a new binary file being saved to the MOC NAS. Binary files are stored in date directories which are inside payload-specific directories. A SOC component called Updater is used to gather all of these files as they are written to the MOC NAS and translate their data frames into ASCII files on the SOC NAS. Updater is usually started from a web front-end as part of the SOC Admin Panel (discussed in a later subsection), but can also be run from the command line. Updater runs as a separate instance for each payload to take advantage of multiple processor machines. Once started, Updater looks for dates listed inside the specified payload directory. Starting at the oldest date directory (or at a user specified date), a payload status file with the current working date is written to the SOC NAS and a chronological list of binary files for that directory is generated (binary filenames are date/time stamps). The oldest file on the list is then opened and data frames are read, processed, and written to the corresponding ASCII files. After one file has been read from the directory, each subsequent file is read until the list is depleted. At this point there is a check to determine if any new files can be added to the list. If so they are added and processed as before. If not, there is a check to see if there is another date directory to read from. If a new date is available, the working directory is changed and the process continues. If not, Updater sleeps for a number of seconds then checks again. This process continues until the program is ended either through the Admin Panel, or manually at the command line.

When a frame is read, a number of things need to happen. In order to understand what is stored in a frame, the frame number must first be read because different data types are transmitted at various cadences. Once the frame number is known, each data type is extracted and stored in memory. For some types of data further processing is needed (e.g. housekeeping values are transmitted as 4-byte integers that need to be multiplied by a full scale factor and sometimes be offset by some amount) and will be performed at this point. Once all the data have been interpreted for a frame it is saved in a number of places. First, the entire frame (except for the medium and slow time resolution spectra) is written to a special file data file that only contains the most recently read data frame for a payload. This file uses the JSON standard[2] and is used by the Live Data Viewer which will be described later. Next, each data group is appended to the end of a day-long CSV ASCII files specific to that group. For example, all the GPS data will be stored in a file named as MMYYDD_GPS.csv and medium time resolution spectral data in MMYYDD_MED.csv.

This description is a bit oversimplified and there is a big caveat that must be discussed. Processing a data file (especially one from a very long MOC call) is not instantaneous. That means if Updater starts to read a file while it is being updated by the MOC it might misinterpret the last frame as bad data although it is really just incomplete. Also, there must be a way to ensure that a file is not determined to be “finished” when really the MOC is still writing it. An obvious solution may be to just stop at the second-to-last file on the list. This would eliminate any chance of reading a file that the MOC is still writing. The problem with this is we want as close to real-time data as possible. With MOC calls lasting over an hour, waiting for the call to terminate is not a viable option. The solution implemented is to first copy the file that is to be read to a local directory and record its name and last modify date to a status file on the SOC NAS (this status file is used by other SOC components as well). Copying the file creates a snapshot allowing the SOC to use what has been written by the MOC without risk of the file changing. After the file is read, its last modify time is checked to see if it has changed. If it has, another snapshot is taken and the new data are extracted. This is repeated until no new data have been written and there are newer files to process.

2. Alert Generation

The Alert Generator runs separately from Updater and handles all active payloads in a single instance (active payloads are set in a common configuration file that can be edited in the Admin Panel). For each active payload, the most recent portion of the ASCII files is read from the SOC NAS and compared to the threshold values stored in JSON files in the SOC NAS' common configuration directory. If any of the data are found to be outside of the threshold value range, the type and time stamp of that data point are written to another file in the common configuration directory to indicate a Yellow Alert. Next, the test for payload transmission timeout is done by finding the most recent file written by the MOC for a given payload and comparing its last modify date to the current system time. If there is a difference that is greater than a predetermined threshold, a code indicating a Red Alert is written to the same file as the Yellow Alerts. Similarly, if a payload is outside of a "safe" latitudes, is at too low an altitude, or has been determined to be sinking too fast, the corresponding Red Alert code is written.

The Red Alert codes will trigger the private SOC to display a message to Mission Monitors indicating a problem. The code must be cleared within a certain amount of time or the SOC will automatically start to warn the team. This is done using the UNIX XMail utility to either email or text message a warning to at least the duty scientist and SOC manager.

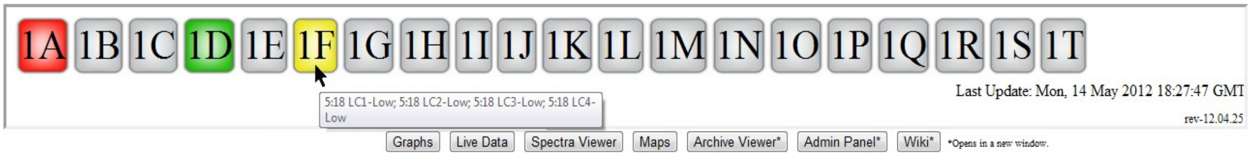
3. Admin Panel

The Admin Panel provides the most important controls for SOC data processing components. It is available on both the Public and Private SOC, but is password protected in either case and is not available to the Mission Monitors. The following controls are provided by the Admin Panel:

1) Start/Stop Updater. This is the front-end for the Updater component. It will allow the user to start and stop the Updater process as well as view system and error logs generated by it.
2) Activate Payload. This edits the common configuration file that determines on which payloads the Alert Generator runs.
3) Threshold Configuration. This edits the JSON files in the SOC NAS common configuration directory that determines when a yellow limit should be triggered. These threshold values also act as Y-axis limits in some plots.

4. Payload Health Bar

Figure 3 shows the Payload Health Bar that always sits atop the SOC display. This is a row of color coded icons representing each payload. The active payload list (as set in the Admin Panel) is read and inactive payloads have their icon changed to gray. Next, the alert file produced by the Alert Generator is read and any active payloads with alerts have their icon changed to the appropriate color (red or yellow). An active payload with no alerts listed results in a green icon.



5:18 LC1-Low; 5:18 LC2-Low; 5:18 LC3-Low; 5:18 LC4-Low

Last Update: Mon, 14 May 2012 18:27:47 GMT

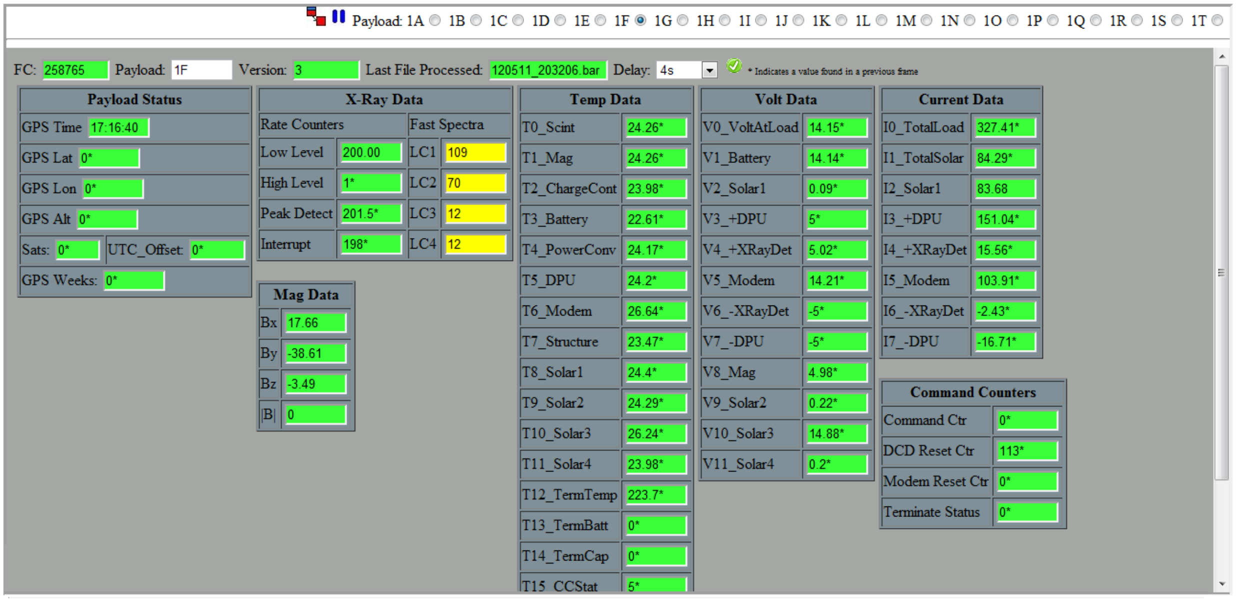
rev-12.04.25

Figure 3. Payload Health Bar. *A color coded icon is shown for each payload indicating data alerts. Mousing over an icon displays a tooltip with the alert information.*

Users can get information about alerts in the form of a tool tip by mousing over the icon. A list with the data type and alert time stamp will be displayed. Furthermore, Red Alerts will trigger a pop up message notifying the Mission Monitor that they must take action. If this message is not cleared in a short amount of time, the SOC will begin contacting the team members. The colors and alert messages are kept up to date by rereading the files from the common configuration directory every 30 seconds using Ajax.

5. Live Data Viewer

The Live Data Viewer is meant to give an easy snapshot view of everything that is happening with a payload. After selecting a payload, the file containing the most recent data frame is read from the SOC NAS. This file is in JSON format and the various data types are simply matched to the field id's. The use of these files makes it easy for the display designer to move fields around or remove them entirely without disturbing the rest of the display.



Payload: 1A ○ 1B ○ 1C ○ 1D ○ 1E ○ 1F ● 1G ○ 1H ○ 1I ○ 1J ○ 1K ○ 1L ○ 1M ○ 1N ○ 1O ○ 1P ○ 1Q ○ 1R ○ 1S ○ 1T ○

FC: 258765 Payload: 1F Version: 3 Last File Processed: 120511_203206.bar Delay: 4s ✓ * Indicates a value found in a previous frame

Payload Status	
GPS Time	17:16:40
GPS Lat	0*
GPS Lon	0*
GPS Alt	0*
Sats: 0*	UTC_Offset: 0*
GPS Weeks:	0*

X-Ray Data			
Rate Counters		Fast Spectra	
Low Level	200.00	LC1	109
High Level	1*	LC2	70
Peak Detect	201.5*	LC3	12
Interrupt	198*	LC4	12

Mag Data	
Bx	17.66
By	-38.61
Bz	-3.49
|B|	0

Temp Data	
T0_Scint	24.26*
T1_Mag	24.26*
T2_ChargeCont	23.98*
T3_Battery	22.61*
T4_PowerConv	24.17*
T5_DPU	24.2*
T6_Modem	26.64*
T7_Structure	23.47*
T8_Solar1	24.4*
T9_Solar2	24.29*
T10_Solar3	26.24*
T11_Solar4	23.98*
T12_TermTemp	223.7*
T13_TermBatt	0*
T14_TermCap	0*
T15_CCStat	5*

Volt Data	
V0_VoltAtLoad	14.15*
V1_Battery	14.14*
V2_Solar1	0.09*
V3_+DPU	5*
V4_+XRayDet	5.02*
V5_Modem	14.21*
V6_-XRayDet	-5*
V7_-DPU	-5*
V8_Mag	4.98*
V9_Solar2	0.22*
V10_Solar3	14.88*
V11_Solar4	0.2*

Current Data	
I0_TotalLoad	327.41*
I1_TotalSolar	84.29*
I2_Solar1	83.68
I3_+DPU	151.04*
I4_+XRayDet	15.56*
I5_Modem	103.91*
I6_-XRayDet	-2.43*
I7_-DPU	-16.71*

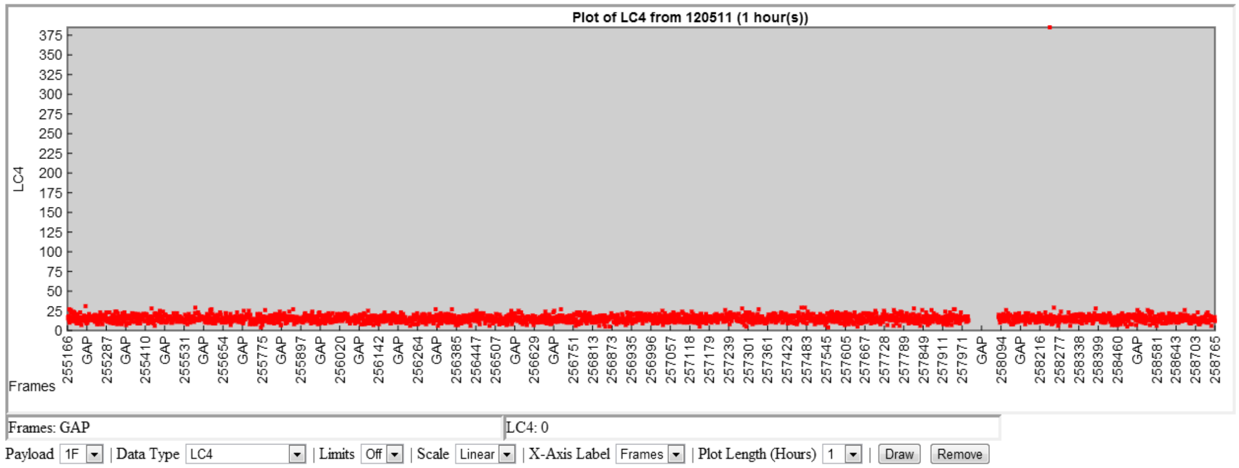
Command Counters	
Command Ctr	0*
DCD Reset Ctr	113*
Modem Reset Ctr	0*
Terminate Status	0*

Figure 4. Live Data Viewer. *The Live Data Viewer gives a quick way to see all data that are transmitted from the payload. These data are checked against threshold values and will change color if out of range.*

The display values are automatically refreshed every four seconds using Ajax with the ability for the user to pause or resume this at any time. Similarly to how the Alert generator works, each value of the displayed data frame is compared to a threshold value and color coded green, yellow, or red as seen in Fig. 4.

6. Quicklook Plots

The Live Data Viewer will give the user an instant look at everything the payload is doing, but it lacks an ability to provide information on history or trends. Quicklook Plots (Fig. 5) displays a simple line chart of the last 1, 6, or 24 hours worth of data returned by the payload. Multiple charts can be added, removed, or refreshed independently,



Frames: GAP | LC4: 0

Payload 1F | Data Type LC4 | Limits Off | Scale Linear | X-Axis Label Frames | Plot Length (Hours) 1 |

Figure 5. Quicklook plots. *These plots show the most recent 1, 6, or 24 hour period of data that have been return for any data type. Any number of plots can be added, removed, or refreshed interdependently.*

again, thanks to Ajax request. The data type, length of time, and other parameters are passed to a server-side program written in PHP which retrieves the most recent data that have been written to the SOC NAS. These requested data are returned as a list of points that Quicklook Plots will display using the new HTML5 Canvas element.

The PHP service must determine where the newest data are located. This is done by reading the current date status file from the payload directory on the SOC NAS. The ASCII file for that data type is then read into memory frame by frame until the requested data set has been created. If the file ends before the needed number of frames have been gathered, the previous day's file is opened and read. This prevents a problem where, for example, requesting a six hour plot at 0100h would result in a one hour plot because the day-long file did not hold sufficient data.

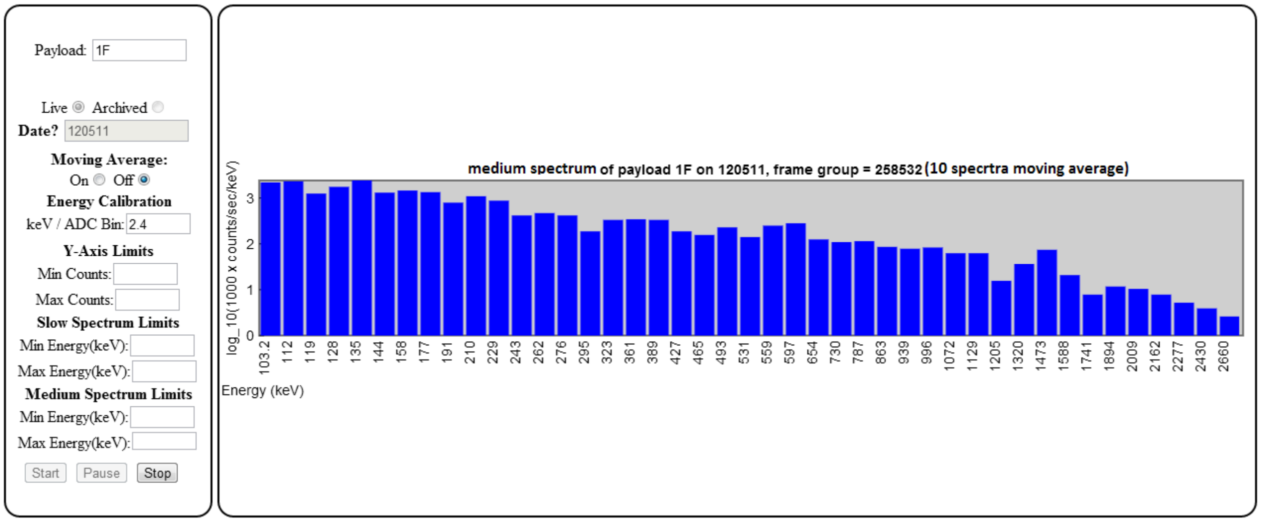
Multiple calls between the MOC and payloads will cause data gaps in the day-long ASCII files. This may result in the retrieval of too much data by the PHP service. For example if one hour of data are requested (3600 frames) during a time period with a 30 minute data gap because communications were down, the returned 3600 frames will actually span 1.5 hours. The solution to this is to expand the data set by filling any gaps with dummy frames. After expansion, however, some of the oldest frames will then need to be removed to create a data set that is of the correct length.

In addition to pulling the desired amount of data, Quicklook Plots also retrieves the Threshold Configuration files stored on the SOC NAS. This configuration file gives an option to have a fixed upper and lower limit to the plots. If these limits are omitted, the plot is simply scaled to fit the data.

The use of the HTML5 Canvas element for drawing plots does put a restriction on the browsers that are able to use this tool. Generally, with the exception of Internet Explorer, modern browsers released after 2010 are able to view charts. IE's most recently released browser should be functional, but is untested with the SOC and will most likely remain unsupported. All charts produced by Quicklook Plots (as well as the Spectrum Viewer and Map) are generated using a custom, open-source JavaScript library, dave.js. Those interested in creating similar charts can find source and distribution packages at http://code.google.com/p/dave-js/.

7. Spectrum Viewer

The previously discussed data displays would be of little use for monitoring spectral data. Figure 6 shows a separate tool that creates histogram charts of both the medium and slow spectra. The current date status file for a user specified payload is used to find the most recent spectra saved to the day-long ASCII files. Histograms can be displayed as either a single spectrum or a moving average of a specified number of transmitted spectra. Again, the data are retrieved using Ajax and automatically refreshed periodically keeping the display current.



Payload: 1F

Live ● Archived ○

Date? 120511

Moving Average:

On ○ Off ●

Energy Calibration

keV / ADC Bin: 2.4

Y-Axis Limits

Min Counts:

Max Counts:

Slow Spectrum Limits

Min Energy(keV):

Max Energy(keV):

Medium Spectrum Limits

Min Energy(keV):

Max Energy(keV):

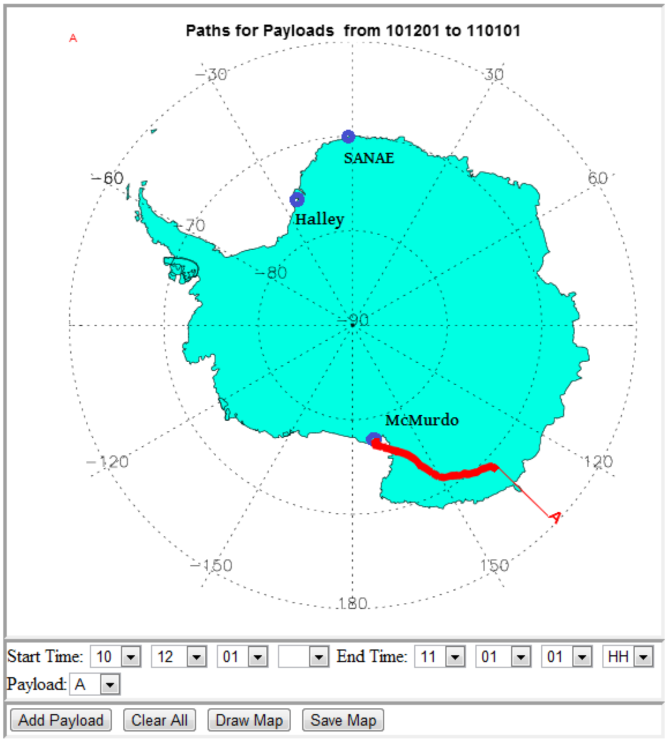
Figure 6. Spectrum Viewer. *X-ray spectra can be viewed in near real time with the spectrum viewer. The user has options to zoom the plot in both the X and Y directions and may take an average spectrum over time.*

The display has a number of options that can be configured. Bin numbers are converted to keV with a user supplied conversion factor. Vertical zoom is provided by allowing the user to set a minimum and maximum for the number of counts displayed. Similarly, horizontal zoom is achieved by limiting which energy levels are displayed.

8. Maps

Using Maps, any number of payloads can be tracked and have their flight path displayed as an overlay on a map of Antarctica as in Fig. 7. Once the user specifies which payloads and time periods to track, Maps requests the needed ASCII data files using the same PHP service as Quicklook Plots. These data then have any points removed that are outside of the specified time range and the map is drawn using dave.js.

This is especially useful when trying to determine the spatial distribution of an event. One could plot the location of the payloads that saw activity during the event in order to find the spread of the precipitation. In addition to this, Maps makes it easy to see the current state of the detector array.



Start Time: 10 12 01 End Time: 11 01 01 HH

Payload: A

Figure 7. Payload Maps. *An example map of a payload's flightpath. The user can select any number of payloads to display during a specific time period.*

Appendix A
Glossary

RBSP Radiation Belt Storm Probes. A mission being launched by NASA in Spring 2012 as part of the Living With a Star program. It aims to study the dynamics of the outer belt.

More information can be found at http://science.nasa.gov/missions/rbsp/

BARREL Balloon Array for RBSP Relativistic Electron Losses. Two campaigns will be flown to support RSBP by detecting electron losses.

More information can be found at http://www.dartmouth.edu/~rmillan/barrel.html

MOC Missions Operations Center. A group of networked hardware that collects and stores raw data from BARREL payloads.

SOC Science Operations Center. A group of networked hardware that processes the MOC data and provides data visualization tools through a web interface.

NAS Network-Attached Storage device. SOC and MOC each use one as a common file system between their networked computers.

Bremsstrahlung Also known as “Braking radiation”, these are X-rays produced by decelerating charged particles that are deflected by another charged particle. This allows the detection of relativistic electron in the atmosphere.

ASCII Files Plain text files that are used to store all data written by the SOC.

CSV Comma Separated Values. Type of ASCII files used for the day-long data files written by the SOC. These files have one record per line and values delimited by commas.

JSON JavaScript Object Notation. Type of ASCII files formatted for easy data transfer between programs (especially useful for reading data into JavaScript applications). http://www.json.org

Ajax Asynchronous JavaScript and XML. Method of requesting data through JavaScript using the XMLHttpRequest object. This creates dynamic websites without the need to reload the entire page.

Acknowledgments

We would like to thank David McGaw, Michael McCarthy, Andrew Liang, Brett Anderson and the rest of the BARREL team for your support in creating the SOC. Also, we would like to thank Manfred Bester from Space Sciences Laboratory at UC Berkely for informing us about SpaceOps and his advice along the way.

References

[1] Millan, R.M. and The BARREL Team, “Understanding Relativistic Electron Losses with BARREL”, *Journal of Atmospheric and Solar-Terrestrial Physics*, Vol. 73, Issues 11-12, July 2011, pp. 1425-1434, doi:10.1016/j.jastp.2011.01.006

[2] Crockford, D., “The application/json Media Type for JavaScript Object Notation (JSON)”, Request for Comments 4627, Network Working Group