REV.	BY	DESCRIPTION	DATE
d	J.R. Hayes	Draft	TBD

Revision: d

Date: October 27, 2014

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
11100 JOHNS HOPKINS ROAD, LAUREL, MARYLAND 20723-6099

EPI-Lo Flight Software Specification

FSCM NO.	SIZE	DRAWING NO.	REV.
88898	A	7464-9005	d
SCALE NONE	DO NOT SCALE PRINT	SHEET 1 OF 181	

Technical Content Approval

Prepared/Approved by:

John R. Hayes
EPI-Lo Software Engineer

Approved by:

Helmut Seifert
EPI-Lo Program Manager

Reid Gurnee
EPI-Lo System Engineer

James Burgum
EPI-Lo Systems Assurance Manager

A Released stamp electronically affixed to the pages of this document certifies that the above personnel or designated alternates have approved this revision. Please refer to the APL Product Lifecycle Management System (PLM) for record of these approvals.

Table of Contents:

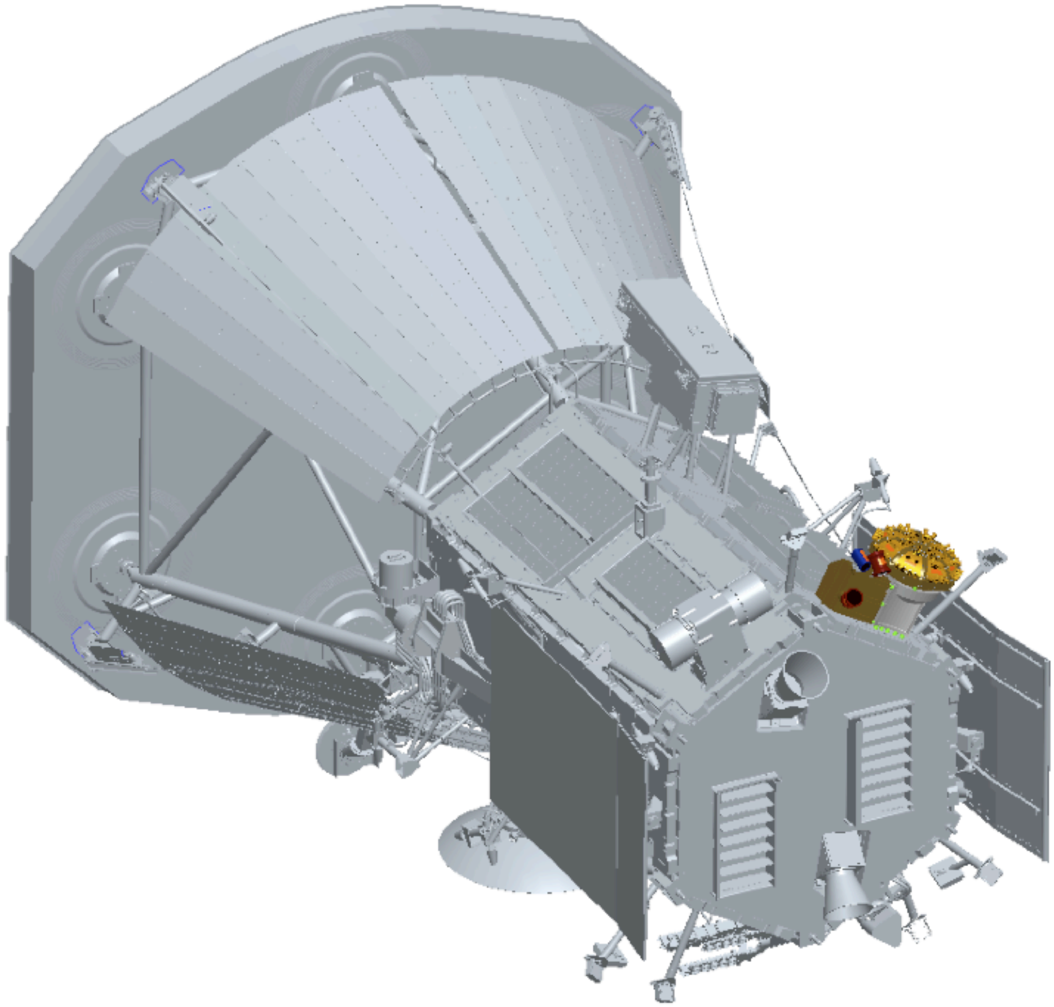
Figures:

Tables:

1. Introduction

The Energetic Particle Instrument - Low Energy (EPI-Lo), part of the Integrated Science Investigation of the Sun (ISIS) suite, is an experiment on the Solar Probe Plus (SPP) spacecraft. The ISIS science objectives are to determine from both gradual (shock accelerated) and impulsive (flare accelerated) solar energetic particle (SEP) events: energy spectra, composition (electrons, protons, major heavy elements) timing, and pitch angle distributions. EPI-Lo is a TOF-based mass spectrometer that measures over 2π steradians the energetic electron (25-500 keV) and ion spectra (~0.02-7 MeV protons & 0.02-2 MeV/nuc heavier ions). EPI-Lo resolves all major heavy ion species and 3He and 4He over much of this energy range in multiple directions. EPI-Lo covers the critical energy range from suprathermal energies (~20 keV/nuc) up to the lower portion of EPI-Hi energy range with a single instrument. ISIS is mounted on the spacecraft as shown in the following figure.

Figure 1. ISIS (EPI-Lo and EPI-Hi) Instrument on SPP Spacecraft

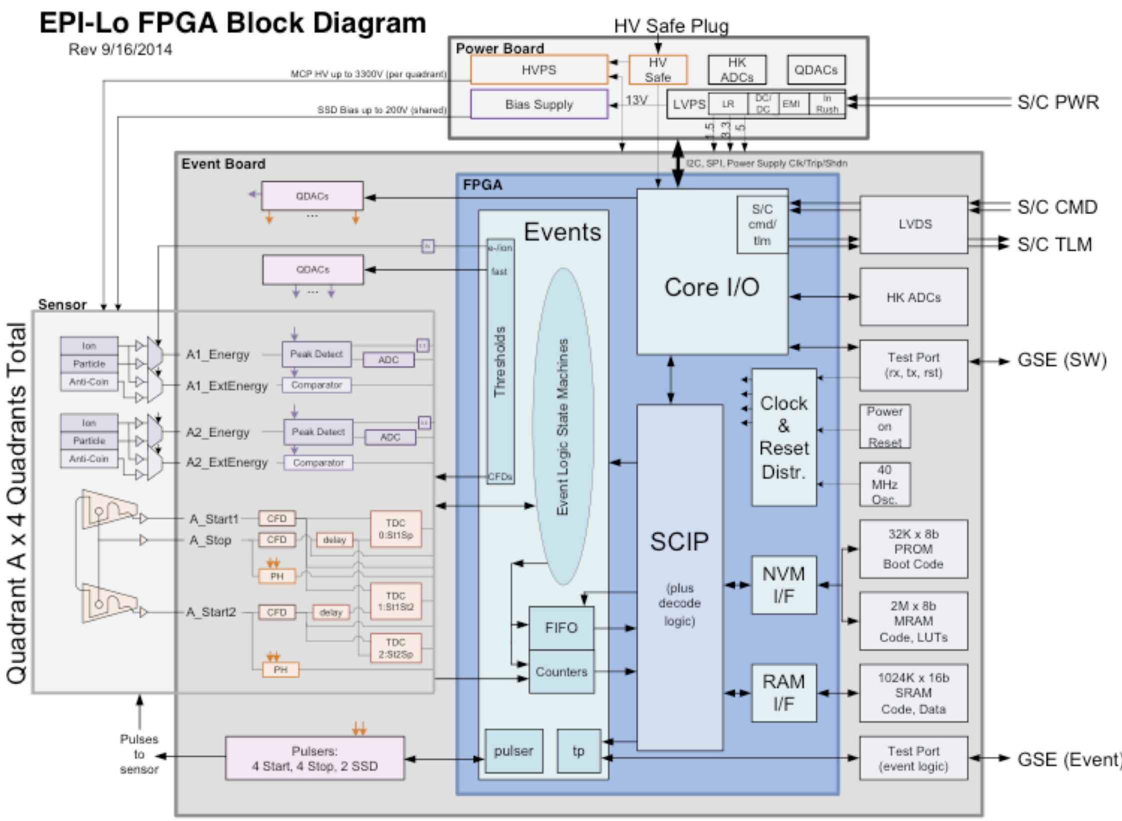


EPI-Lo is comprised of two subsystems, the sensor and the main electronics. The sensor consists of eight wedges, each having 10 apertures and look directions. The electronics are separated into two boards, the event board and the power supply board. The sensor head and main electronics are mechanically integrated together and mounted as a single unit to the spacecraft. Each sensor wedge includes a time-of-flight (TOF) section feeding a solid-state silicon detector (SSD). The SSD and its associated preamps connected to the event board measures particle energy. Secondary electrons, generated by ions passing through the entry and exit foils, are detected by the timing anodes and their associated preamps to measure ion TOF. Event energy and TOF measurements are combined to derive ion mass and to identify particle species.

The following figure shows the EPI-Lo electronics block diagram. On the left, the sensor generates analog representations of particle Time Of Flight (TOF) and energy.

The event board digitizes the TOF and energy and reads the events into a Field Programmable Gate Array (FPGA). The FPGA contains event processing logic and a processor. Some events are passed to software running on the processor for further analysis and science processing. The event board also includes a spacecraft command and telemetry interface. The power supply board has housekeeping electronics and a Low Voltage Power Supply (LVPS). It also has a Bias Voltage (BV) power supply for the SSDs and a High Voltage (HV) power supply.

Figure 2. EPI-Lo Block Diagram



This document describes the EPI-Lo flight software from a user's point-of-view, including science data processing, commands, and telemetry. The document deals primarily with the application software; however, the boot software is described near the end. Additional information can be found in the references. Reference 1 documents the software requirements and includes compliance matrices. This specification summarizes the SPP spacecraft interface; the full interface is defined in reference 2.

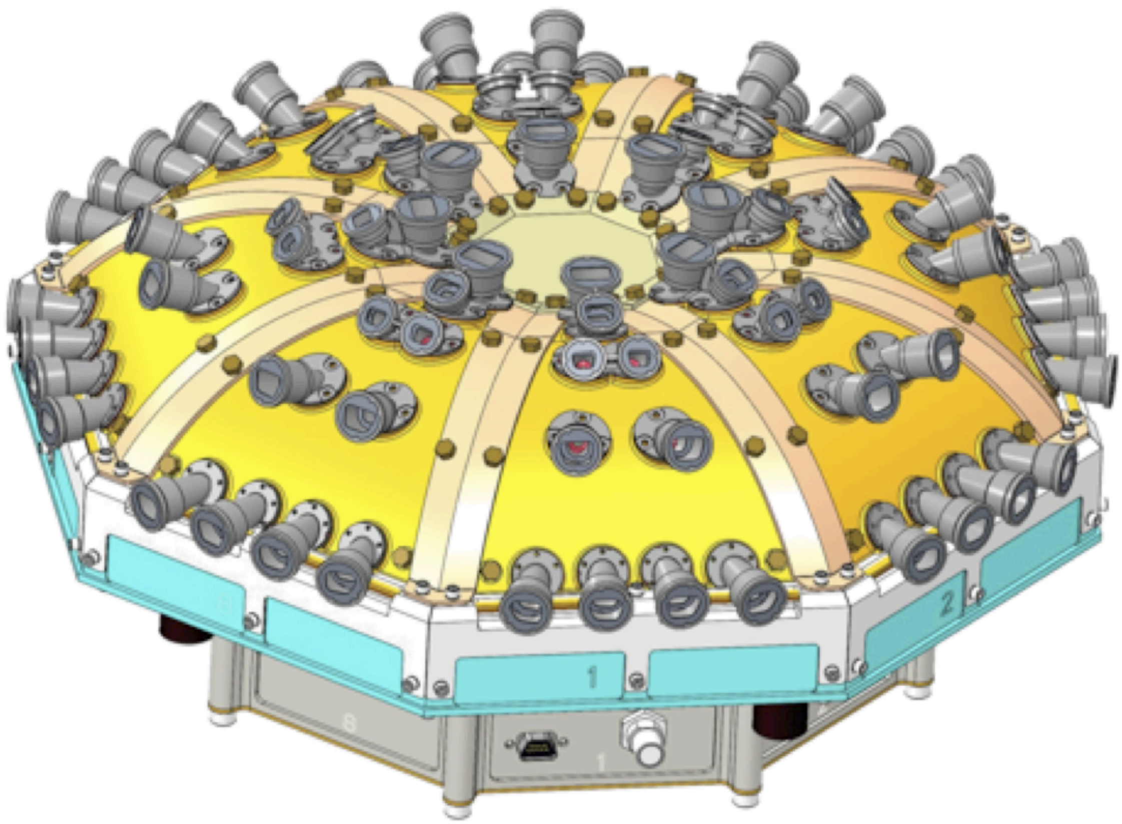
2. Event Processing

EPI-Lo analyzes ions and electrons. The sensor and analysis electronics are described below.

2.1 Sensor

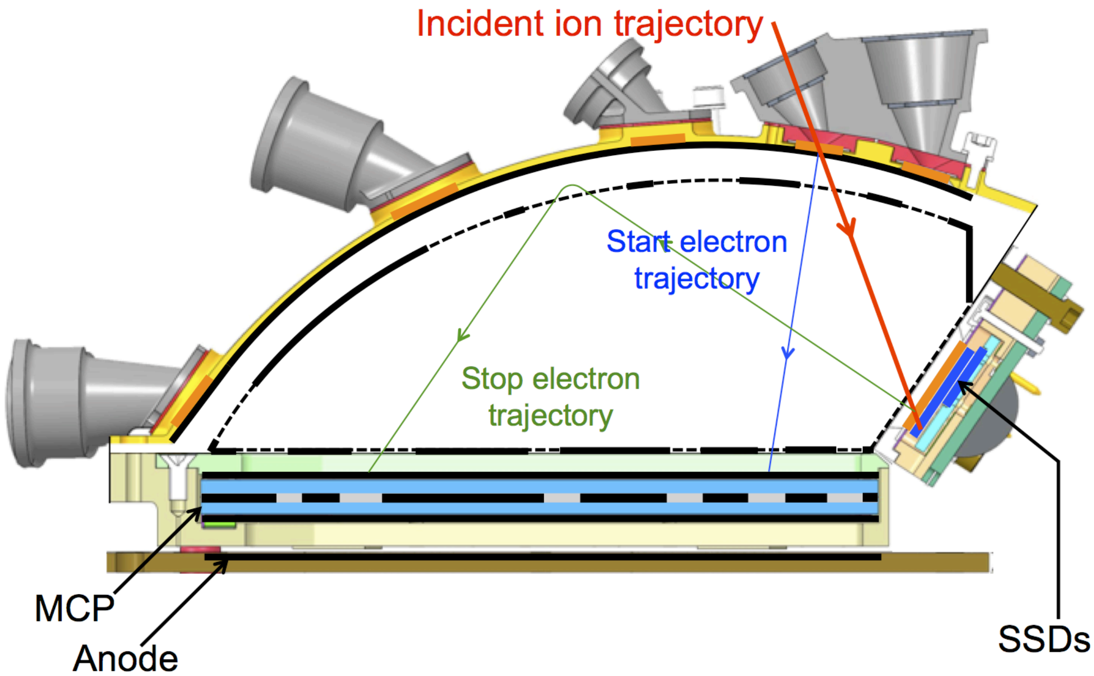
The EPI-Lo instrument has an array of sensors that views a solid angle of 2π steradians. There are eight wedges and each wedge has ten apertures giving a total of 80 look directions.

Figure 3. EPI-Lo Sensors



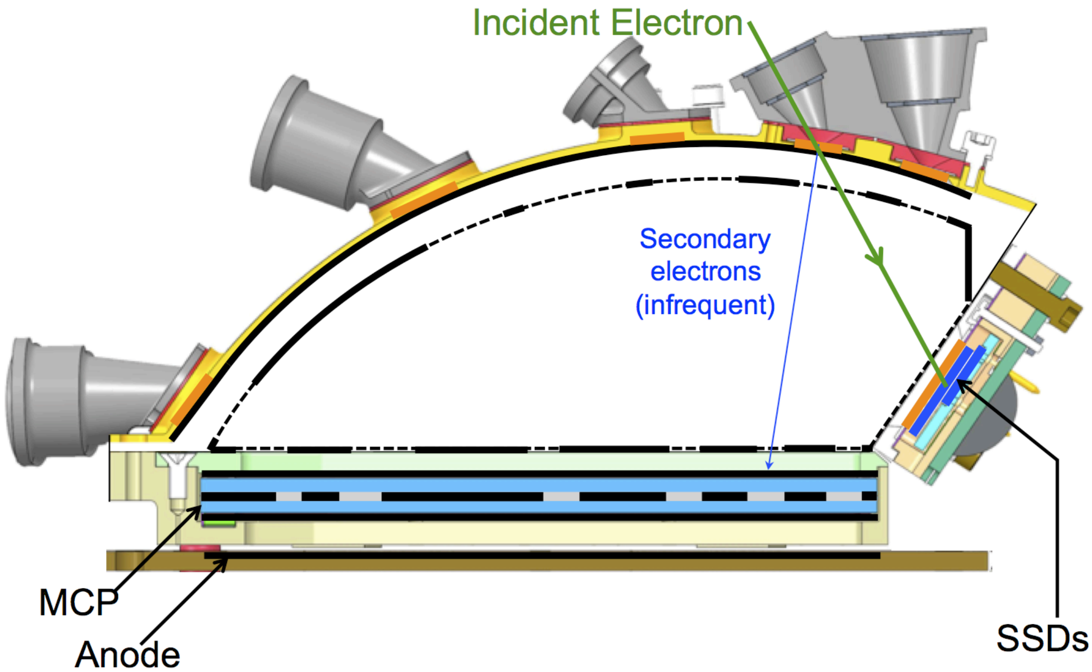
The instrument includes Solid State Detectors (SSD) that measure particle energy and Micro-Channel Plate (MCP) detectors that measure particle Time Of Flight (TOF) and position. An ion entering the instrument, as shown in the following figure, will strike the SSD generating an energy measurement. As the ion passes through the aperture it hits a start foil, dislodging electrons. The electrons are accelerated towards the MCP, which amplifies the signal. The electrons land on a start anode generating a start pulse. Similarly, just before reaching the SSD, the ion hits a stop foil. The resulting electrons are also accelerated towards the MCP and a separate anode generates a stop pulse.

Figure 4. EPI-Lo Sensor - Ion Detection



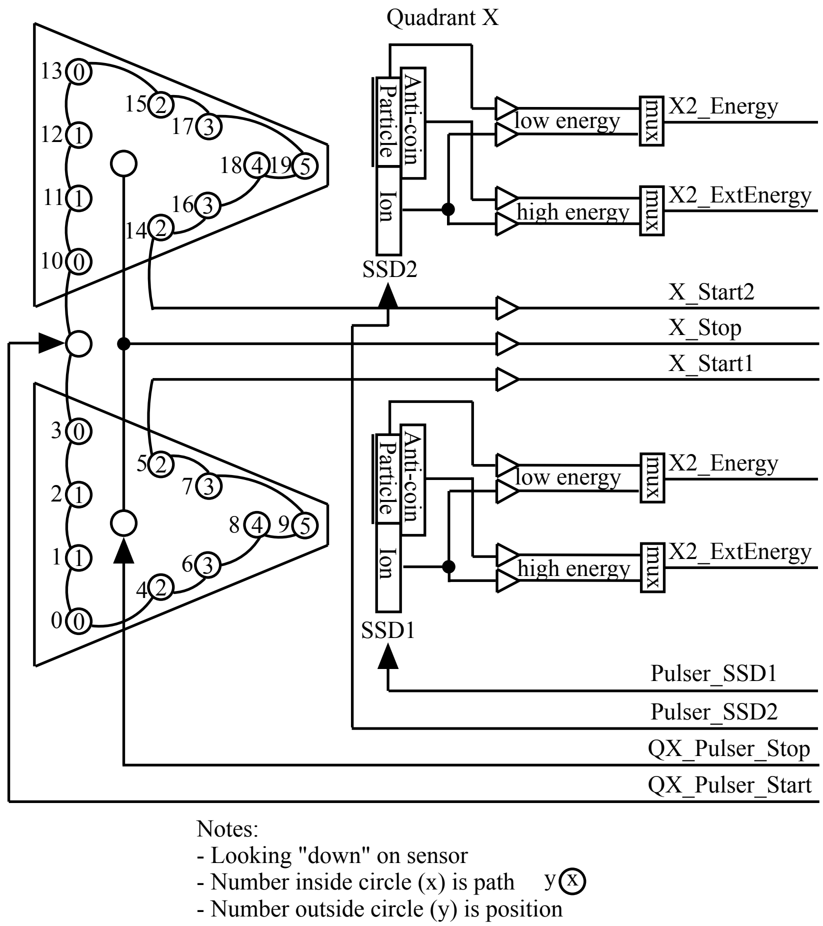
The SSDs can also measure the energy of electrons that enter the instrument. A small fraction of electrons will dislodge secondary electrons from the start foil to provide a measurement of the aperture position; no TOF measurement is made.

Figure 5. EPI-Lo Sensor - Electron Detection



Each of the eight wedges include anodes and SSDs with associated electronics. However, the anodes in each pair of wedges are connected and share TOF measurement electronics. Therefore, pairs of wedges are grouped together to form quadrants; each quadrant operates independently. The following figure shows one EPI-Lo sensor quadrant consisting of two wedges. Each wedge has ten start anodes and a stop anode. The start anodes are chained together. The start anode chain in one wedge is connected to the start anode chain in the other wedge in the quadrant. The stop anodes of the two wedges are also connected together. The start pulse will appear at both ends of the chain, i.e. a start1 and a start2 pulse. The time between the start pulses provides a measurement of the position of the aperture; between the two wedges there are twenty valid positions per quadrant. The times between the start pulses and the stop pulse provide redundant measurements of the TOF.

Figure 6. EPI-Lo Sensor Block Diagram



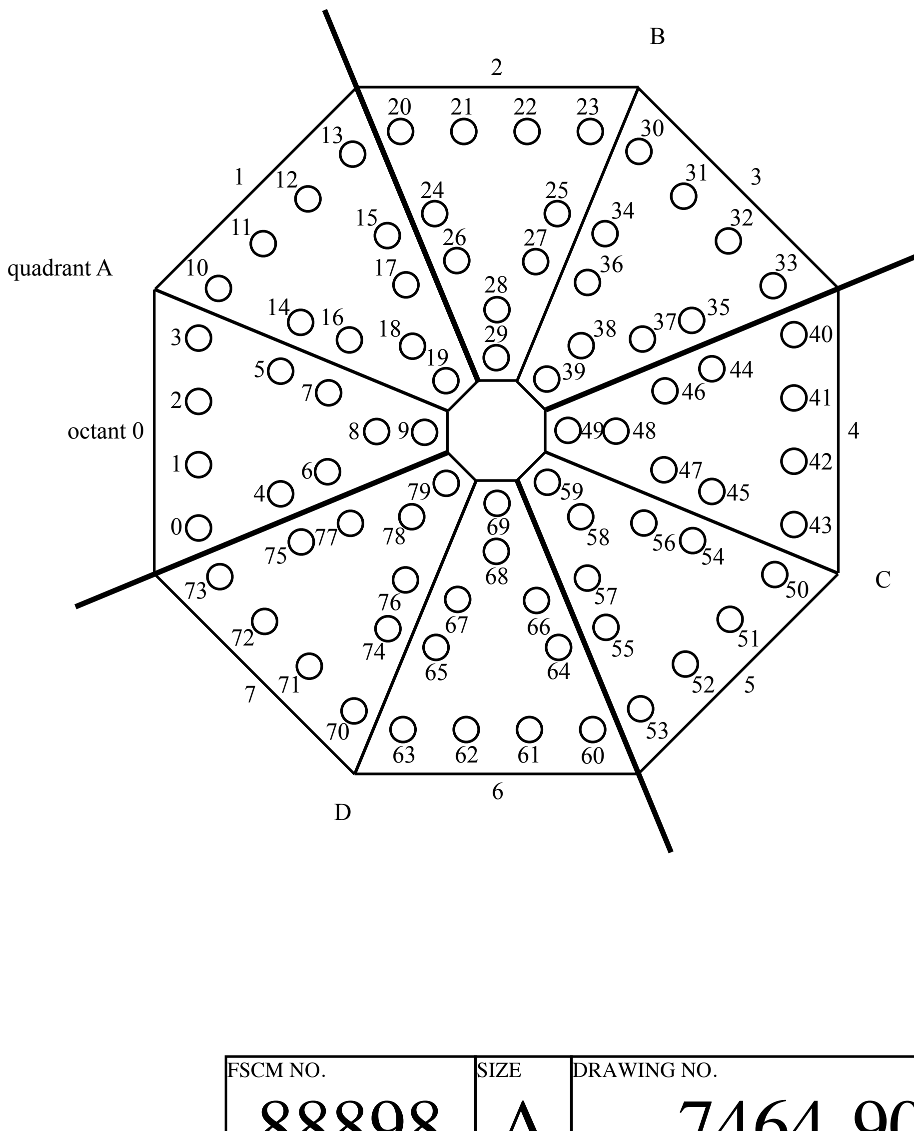
Each wedge has two SSDs, one for ions and one for particles, i.e. ions and electrons, as shown in the figure above. The particle detectors differ from the ion detectors in that there is an added layer of aluminum, which excludes low-energy ions. However, the particle detectors can be used to measure high-energy ions. Each energy measurement has an associated extended energy measurement. For the particle SSDs, this will be the energy detected in an anti-coincidence detector mounted behind the SSD. The anti-coincidence detector allows EPI-Lo to detect and discard unwanted background events. Alternatively, the anti-coincidence detector allows EPI-Lo to detect high-energy ions using the particle detector. For the ion SSDs, the extended energy measurement provides, a low-gain, i.e. higher-energy range, measurement of the ion energy.

There is only one set of energy measurement electronics per wedge. Therefore, the ion and particle SSDs cannot be used simultaneously. To collect data from both SSDs the electronics is time-multiplexed between them (see below).

The figure above shows signals from an internal pulser injected into the start anodes, the stop anodes, and the two SSDs (see 4.2). Note that a pulse injected into the anode system will have its own “position”.

The quadrants are labeled A, B, C, and D. From the quadrant and position (0 - 19) determined by the hardware, the EPI-Lo software calculates a look direction (0 - 79) that covers the full field of view. The look direction is used in the accumulation of high spatial resolution data products. For events with no position value, the SSD is used instead. From the octant and the SSD, the EPI-Lo software calculates an octant number (0 - 7). The octant is used in the accumulation of low spatial resolution data products. The quadrant, octant, and look direction labeling scheme is shown in the following figure; this is the “downward” view, i.e. into the sensors. The two SSDs in each quadrant are labeled 1 and 2; for example, the quadrant B SSDs are named B1 and B2. (This is not shown in the figure.)

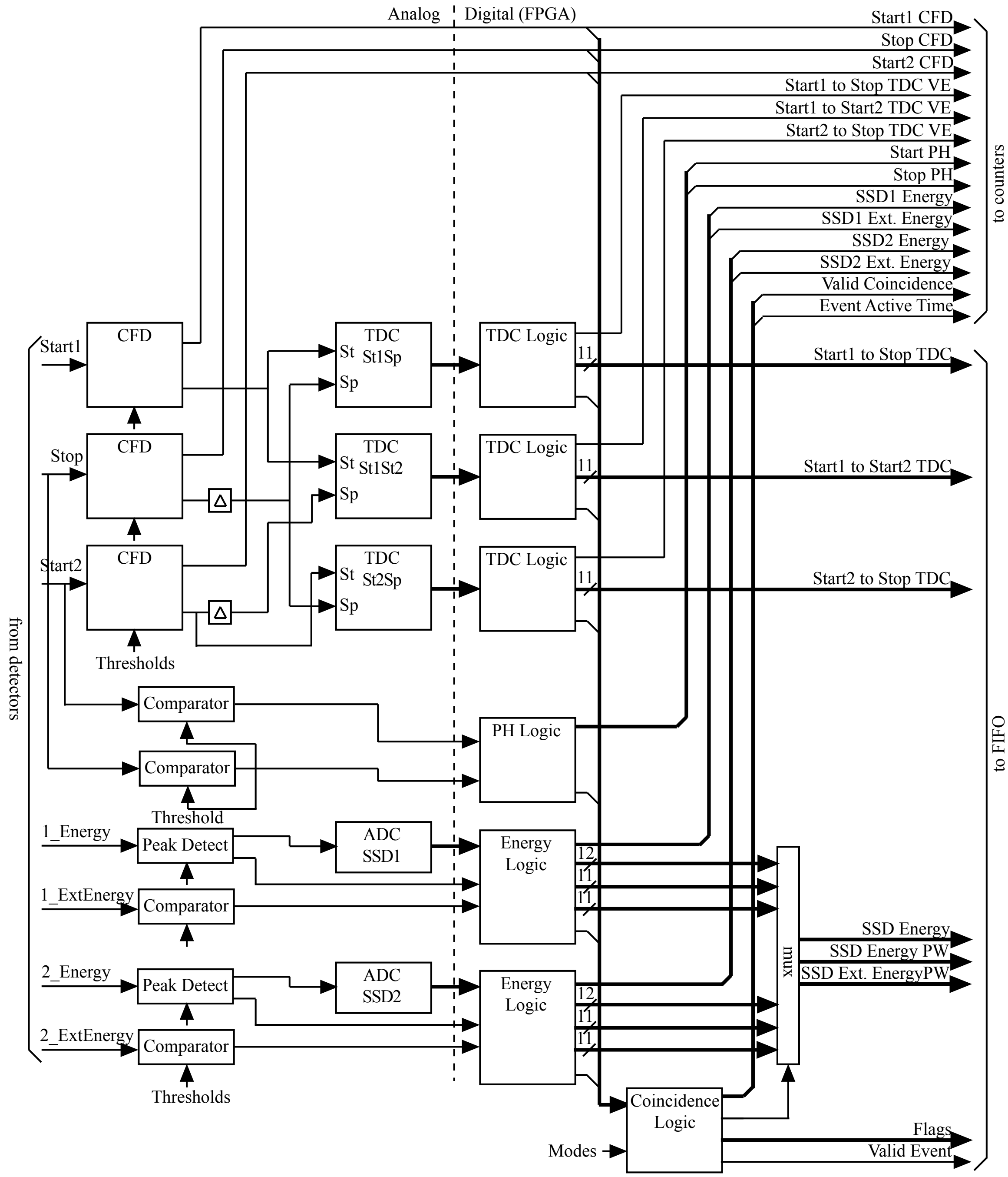
Figure 7. EPI-Lo Quadrant, Octant, and Look Direction Labeling



2.2 Analysis

The following figure shows how the pulses from a single sensor quadrant are analyzed in the event board. All four sensors have identical hardware and can be configured independently. Analysis begins in analog circuitry and continues in digital logic within the FPGA. Pulses from the MCP anodes go to Constant Fraction Discriminators (CFDs). A CFD triggers on a constant fraction of the total peak height and thus generates a timing pulse independent of the height of the input pulse. Each CFD has a commandable threshold (EPILO_TDC_CFD). Pairs of CFD output pulses go to start/stop inputs to Time/Digital Convertors (TDCs), which measure the time between the pulses. In this case these are the Time Of Flights (TOF) Start1 to Stop and Start2 to Stop and position Start1 to Start2. Logic in the FPGA further analyzes event data from each TDC. The TDC logic forwards a time value and signals for coincidence tests (see below).

Figure 8. Pulse Analysis



The start2 and stop pulse heights (PHs) are compared against commandable (EPILO_TDC_PH) thresholds; the thresholds can also be stepped by command (EPILO_TDC_PH_STEP). The results of the comparison are available in flag bits. Note: these are not used for coincidence tests.

The two SSD energy pulses are measured by peak detect circuitry and an Analog/Digital Converter (ADC). Each peak detect has a commandable threshold (EGY_ENERGY); there are separate threshold settings for the particle vs. the ion SSDs. Logic in the FPGA further analyzes energy data. The width of the energy pulse is measured; this is used to extend the range of energies that can be analyzed. The extended energy pulse width is also measured by comparing the pulse against a commandable threshold (EGY_EXTEND). Again, there are separate threshold settings for particles vs. ions. For ions, the extended energy pulse width extends the energy range beyond what is provided by just the energy pulse width. For particles, it can be used for anti-coincidence or extended energy measurements. The energy logic generates SSD1 and SSD2 energy values and signals for coincidence tests (see below). The coincidence logic chooses whether to forward the SSD1 or the SSD2 values.

Most of the individual pulses are counted (see 2.4). The measured TDC and energy values can be placed in a First-In First-Out (FIFO) for further analysis by the software (see 2.3). Whether the measured event data is placed in the FIFO is determined by the event mode. Events are processed in one of four possible modes. There are modes for measuring ion composition, ion energy, particle composition, and particle energy. There are separate ion and particle SSDs. The event mode selects which detector to use. Ion composition mode is triggered by the TDC subsystem and generates TOF, position, and ion energy measurements; a TDC-only option, i.e. with no energy, can be selected (see below). Ion energy mode measures ion energy only. Particle composition mode is triggered by the TOF subsystem and generates TOF, position, and particle energy measurements. Particle energy mode measures particle energy, but can measure position for some events.

Table 1. Event Modes

Mode	SSD	Description
Ion Composition	Ion energy and pulse-width plus high-energy pulse-width	TDC-triggered; measures TOF, position, and energy; produces ion composition and TOF histograms
Ion Energy		Energy-triggered; measures energy only; produces ion energy spectra
Particle Composition	Particle energy and pulse-width plus anti-coincidence pulse-width	TDC-triggered; measures TOF, position, and energy; produces composition of moderate energy ions and TOF histograms
Particle Energy		Energy-triggered; measures energy (and some positions); produces electron energy spectra and composition of high energy ions

Whether or not events that generate pulses in both SSD channels are accepted or rejected can be set by command (EPILO_EVT_MULTI). If the event is accepted, the SSD to use will be selected randomly. This option is used in all event modes. There are commands that refine what constitutes a valid event in ion and particle composition modes. Whether or not the position calculated must be valid can be commanded (EPILO_EVT_TDC_POS). Whether or not the position calculated must match the SSD channel can also be commanded (EPILO_EVT_TDC_MATCH). Whether or not the TDC data is internally consistent, i.e. Start1Stop + Delay - Start2Stop ~= Start1Start2, can be commanded (EPILO_EVT_TDC_CHECK). Events with no SSD energy measurement can be enabled, decimated, or disabled by command (EPILO_EVT_TDC_ONLY).

The FPGA can only process events in one mode at a time. Further, the mode determines whether the ion or the particle SSD is selected. To collect both particle and ion events, the hardware is time-multiplexed between the different modes. This tries to present the illusion of independent particle and ion electronics. So, energy subsystem commands and parameters have both particle or ion settings. For example, there are commands to set the particle energy threshold and to set the ion energy threshold. The time-multiplexing loads the actual hardware threshold with either the particle and ion setting depending on the event mode.

2.3 Event FIFO

The EPI-Lo hardware passes valid particle event data to the software for further analysis. The events pass through a First-In First-Out (FIFO). The FIFO contents are listed in the following table, grouped into TDC values, energy values, and quadrant number. In ion and particle composition modes, both the TDC and energy values can be used. In ion energy mode, only the energy values can be used. In particle energy mode, the energy values can be used, and possibly the position from the TDC values. The quadrant number can be used in any mode.

Table 2. Event Data

Name	Bits	Description

Start1 to Stop TDC	11	TOF
Start1 to Start2 TDC	11	Position
Start2 to Stop TDC	11	TOF
TDC Valid Flags	3	TDC validity flags
MCP Coin Flags	2	MCP PH flags (Start2 and Stop)
TOF	11	Corrected TOF, average of two TOF values above
Position	5	Calculated position 0-19, 20 = internal pulser, 31 = unknown
SSD Energy	12	SSD energy
SSD Energy PW	11	SSD energy pulse width (25ns units)
Ext. Energy PW	11	Extended energy pulse width (25ns units)
SSD Coin Flags	5	SSD with anti, start1, start2, stop, or TDC VEs
SSD Chan	2	SSD channel 0 = SSD1, 1 = SSD2, 2-3 = invalid
Quadrant	2	Quadrant 0 = A, 1 = B, 2 = C, 3 = D

Start1 to Stop and Start2 to Stop provide redundant measurements of the TOF value. TOF provides the corrected TOF value, i.e. the average of the two TOF values. Start1 to Start2 provides a position measurement. This value is compared against an array of thresholds to determine a position number of 0 to 19, or a position corresponding to an injected internal pulse. The thresholds are uploadable parameters. The TDC Valid Flags indicate which TDCs have valid data. The MCP PH Flags indicate whether Start2 or Stop had pulses exceeding their commanded thresholds.

SSD Energy is the value of the selected energy channel, SSD1 or SSD2. The channel selected is indicated in SSD Chan. A baseline value is subtracted from the measured energy. There are separate baselines for each particle and ion SSD; these are uploadable parameters. The SSD Coin Flags indicated whether there was a coincident anti-coincidence, start1, start2, or stop pulse or TDC VE pulses. Besides the SSD energy itself, there is an energy pulse width measurement and an extended energy pulse width measurement.

The FIFO is 64 events deep. If the EPI-Lo software is unable to read all of the events from the FIFO, the FIFO will fill and subsequent events will be discarded.

2.4 Event Counters

The EPI-Lo hardware counts a variety of pulses from the sensor. CFD pulses, TDC valid events, PH pulses, and energy pulses are counted. The number of valid coincident events and the total activity time of the event analysis logic is counted. The hardware counters are 21 bits. Identical sets of counters are maintained for quadrants A, B, C, and D. Finally, the total number of events written from all four quadrants to the FIFO are counted.

Table 3. Hardware Event Counters

Name	Description
A Start1 CFD	A CFD pulses
A Start2 CFD	
A Stop CFD	
A Start1 to Stop TDC Valid	A TDC valid events
A Start2 to Stop TDC Valid	
A Start1 to Start2 TDC Valid	
A Start PH	A PH pulses
A Stop PH	
A1 SSD Energy LED	A energy pulses
A1 Extended Energy	
A2 SSD Energy LED	
A2 Extended Energy	
A Valid Coincidence	A valid coincidence events
A Event Active Time	A event analysis activity time
B Start1 CFD	B pulses
...	
B Event Active Time	
C Start1 CFD	C pulses
...	
C Event Active Time	
D Start1 CFD	D pulses
...	
D Event Active Time	
FIFO Write	FIFO writes

3. Science Data Processing

The following sections describe EPI-Lo science data processing, from low-level event processing to integration of the science data products.

3.1 Data Accumulation

The EPI-Lo software divides each second into eight evenly spaced time slots. The slots are numbered 0 through 7. This is illustrated in the following figure.

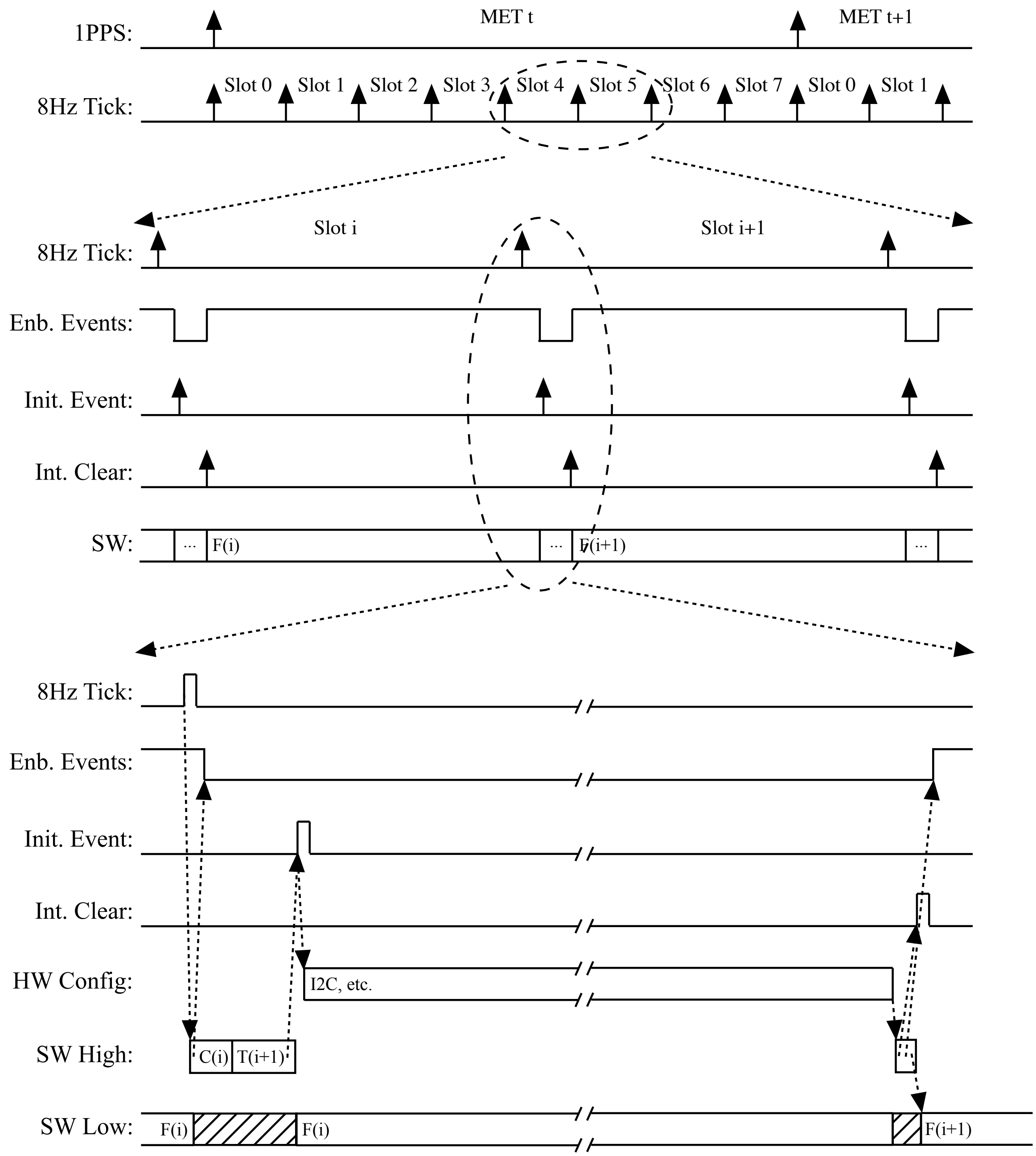
Figure 9. Slot Schedule

Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7

The sensor hardware can be placed in a different event mode during each slot. The dark bars in the figure represent a fixed dead-time, ~4.1 ms, for switching between modes. The pattern of modes in each slot is commandable (EPILO_DAT_COLLECT). Any slot may collect data in any mode. Each pattern collects different data in different proportions. For example, setting slots 0 through 4 to ion composition mode and slots 5 through 7 to particle energy mode collects ion composition events 5/8's of the time and particle energy events the rest of the time; ion energy and particle composition events are not collected at all. Note: if two adjacent slots have the same mode, there will still be a dead-time between the slots.

The EPI-Lo hardware accumulates counts and events for one slot. Every slot, the software stops the integration, reads out the results, and starts a new accumulation. The timing is summarized in the following figure. The start of slot 0 is synchronized to the 1PPS received from the SPP spacecraft. At the start of each slot, the software turns off accumulation: the counters are frozen and all incoming events are discarded. The counters are not double-buffered; the software reads them into memory. The software commands the hardware to zero the counters and empty the raw event FIFO, and then, turns accumulation back on. After the previous slot's counts are read out, the software starts reading raw events corresponding to the next slot.

Figure 10. Accumulation Timing



3.2 Data Product Integration

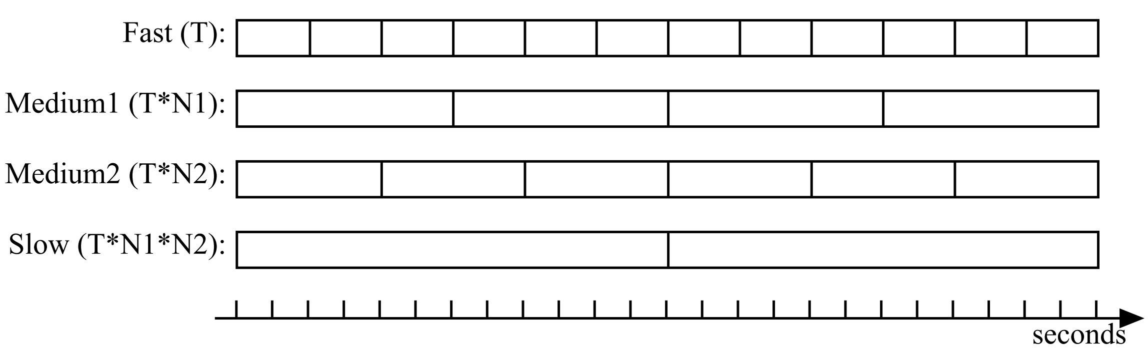
The data products generated by the EPI-Lo software depend on the event modes commanded in the slot schedule. The following table lists the products generated in each mode; the precise definition of these products appears in subsequent sections.

Table 4. Data Products Per Event Mode

Mode	Data Products
Ion Composition	• Basic and diagnostic rates • High-resolution protons, high time-resolution protons, group1 ions, and group2 ions • PH scan, mass histograms, and ion TOF • Raw and priority events
Ion Energy	• Basic and diagnostic rates • Ion energy • Raw events
Particle Composition	• Basic and diagnostic rates • Particles and particle TOF • Raw and priority events
Particle Energy	• Basic and diagnostic rates • High-resolution electrons, high time-resolution electrons, and high look-resolution electrons • High-energy particles • Raw and priority events

The EPI-Lo software reads data from the hardware for further integration into science data products. The data products can be grouped depending on their integration cadence, fast, medium, or slow. The integration time is set by command (EPILO_DAT_TIME). The command has three arguments, T, N1, and N2. T specifies the number of seconds to integrate fast data products. N1 specifies the number of fast integrations that make up a medium1 integration; in other words, medium1 data products are integrated for T*N1 seconds. Similarly, N2 specifies the number of fast integrations that make up a medium2 integration; in other words, medium2 data products are integrated for T*N2 seconds. Slow data products are integrated for N1*N2 fast integrations, i.e. slow products are integrated for T*N1*N2 seconds. The following figure shows an example integration, with T=2, N1=3, and N2=2.

Figure 11. Data Product Integration Example (T=2, N1=3, N2=2)



The following table lists the EPI-Lo data products, with their classification of fast, medium, or slow. The table lists default cadences; the actual cadences can be modified via uploadable parameters. The precise definition of these products appears in subsequent sections.

Table 5. Default Data Product Integration Times

Cadence	Data Products	Integration Time
Fast	• PH scan • Particle energy basic rates • High time-resolution electrons • All raw events • All priority events	T
Medium1	• Ion composition basic rates • High time-resolution protons • Ion energy basic rates • Particle composition basic rates	T*N1
Medium2	• Ion composition diagnostic rates • Ions group1 • Ion TOF • Ion energy diagnostic rates • Particle composition diagnostic rates • Particle TOF • Particle energy diagnostic rates • High look-resolution electrons • High-energy particles	T*N2
Slow	• High-resolution protons • Ions group2 • Mass histograms • Ion energy • Particles • High-resolution electrons	T*N1*N2

Most of the data products can be enabled or disabled by command. Most products are explicitly commanded (EPILO_DAT_ENB); raw and priority event products are disabled by setting the number of events to zero (EPILO_DAT_EVT). Data products will also be disabled automatically if they are not collected (EPILO_DAT_COLLECT). For example, if ion energy mode is not scheduled in any of the slots, then Ion Energy Basic Rates will not be produced, regardless of its commanded enable or disable state.

Event analysis varies with the hardware event mode, i.e. ion composition, ion energy, particle composition, and particle energy. In ion composition mode, rate box counts corresponding to groups of species and energy are generated (see 3.3). In ion energy, a histogram of the ion energy only is generated (see 3.4). Particle composition mode is really a moderate-energy ion composition mode and works similarly (see 3.5). In particle energy mode, histograms of the electron energy are generated as well as high-energy ion composition (see 3.6).

3.3 Ion Composition

EPI-Lo produces ion composition and TOF histograms in ion composition mode. The following two figures summarize the data flow from hardware counts and events to telemetry packets. The packets can be categorized into rates, binned data, and single event data; these are described below. In the figures, boxes denote data and ovals computations (see key in first figure).

Figure 12. Ion Composition Dataflow: Part 1

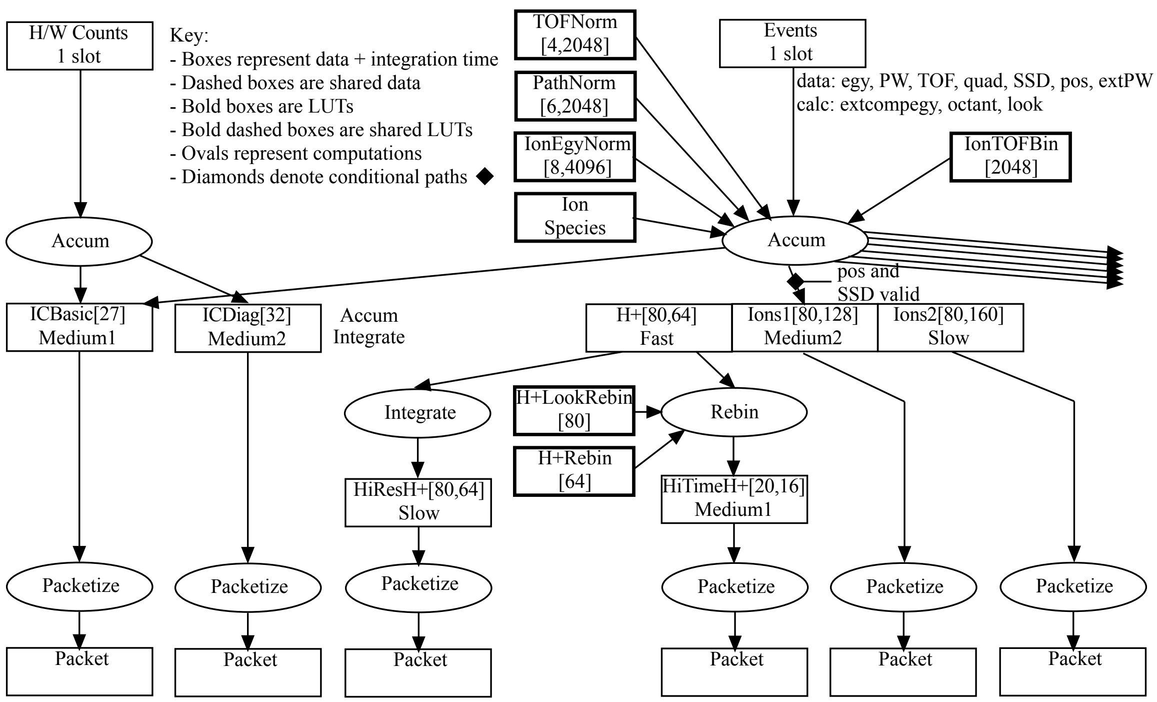
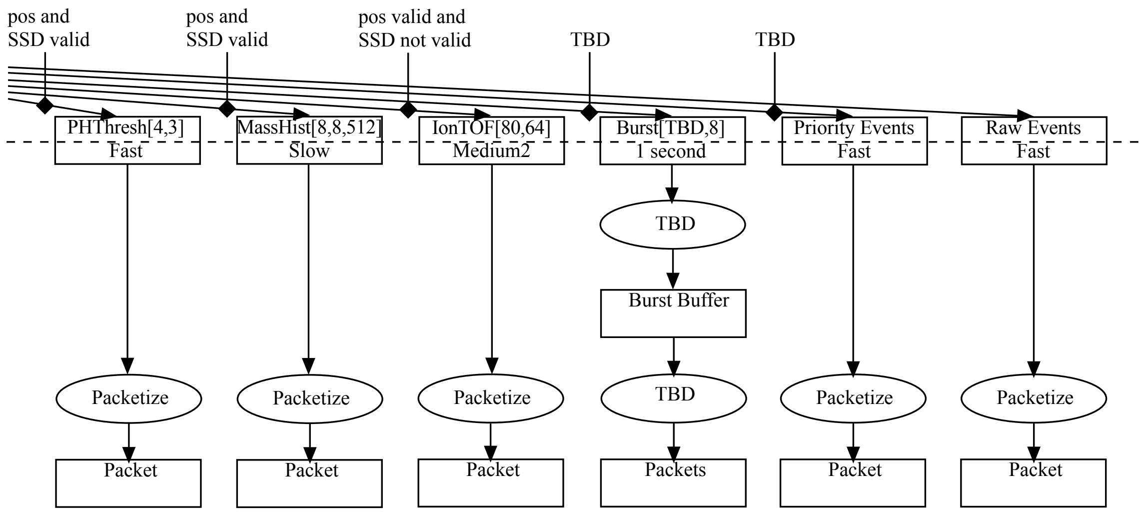


Figure 13. Ion Composition Dataflow: Part 2



3.3.1 Rate Counters

The EPI-Lo software integrates the hardware event counters into a basic and diagnostic rates data products. The 21-bit hardware counters are read once per slot and accumulated in 32-bit counters in software. The software also counts the number of pulses it determines are from the internal pulsers and the total number of events it is able to process. The key hardware counters and both software counters comprise the basic rates. Additional hardware counters comprise the diagnostic rates. These are shown in the following two tables.

Table 6. Ion Composition Basic Counters

Name	Description
A Start1 CFD	A CFD pulses
A Stop CFD	
A1 SSD Energy LED	A energy pulses
A2 SSD Energy LED	
A Valid Coincidence	A valid coincidence events
A Event Active Time	A event analysis activity time
...	B pulses
...	C pulses
...	D pulses
FIFO Write	FIFO writes
Internal Pulses	Internal pulse events
Processed	Events processed by software

Table 7. Ion Composition Diagnostic Counters

Name	Description
A Start2 CFD	A CFD pulses
A Start1 to Stop TDC VE	A TOF chip VE pulses
A Start2 to Stop TDC VE	
A Start1 to Start2 TDC VE	
A Start PH	A PH pulses
A Stop PH	
A1 Extended Energy	A energy pulses
A2 Extended Energy	
...	B pulses
...	C pulses
...	D pulses

3.3.2 Binned Event Data

The EPI-Lo software analyzes ion composition event data. Each event read from the FIFO includes a position. If this is one of the twenty valid positions, the event will be binned as described below. If the position indicates that the event was generated by EPI-Lo itself, i.e., a test pulse event, the event is counted in the rates data. This excludes test pulses from the binned event data. Regardless of the position, the event is a candidate for being saved as raw data and will be counted as a processed event in the rates data. This is summarized in the following pseudo-code.

```
accum-ion-comp-event:
    -- Precondition: event ready: egy, PW, TOF, quad, SSD, pos, extPW
    if pos valid                -- i.e. pos {0..19}
        look = 20*quad + pos
        ionTOF = PathNorm[Path[pos], TOFNorm[quad, TOF]]
        if SSD valid            -- i.e. not TDC-only
            octant = 2*quad + SSD
            -- bin protons and ions
            ionEgy = IonEgyNorm[octant, extCompEgy(egy, PW, extPW)]
            bin = IonSpecies[ionEgy, ionTOF]
TBD: add a "junk" bin that is discarded?
            if bin < #protons   -- if proton
                H+[look, bin]++
            else if bin < (#protons+#ion1s)
                Ions1[look, bin - #protons]++
            else
                Ions2[look, bin - (#protons+#ion1s)]++
            -- some TBD ions are binned in PH threshold scan
            if bin TBD
                if Start-PH bit
                    PHThresh[quad, 0]++
                if Stop-PH bit
                    PHThresh[quad, 1]++
```

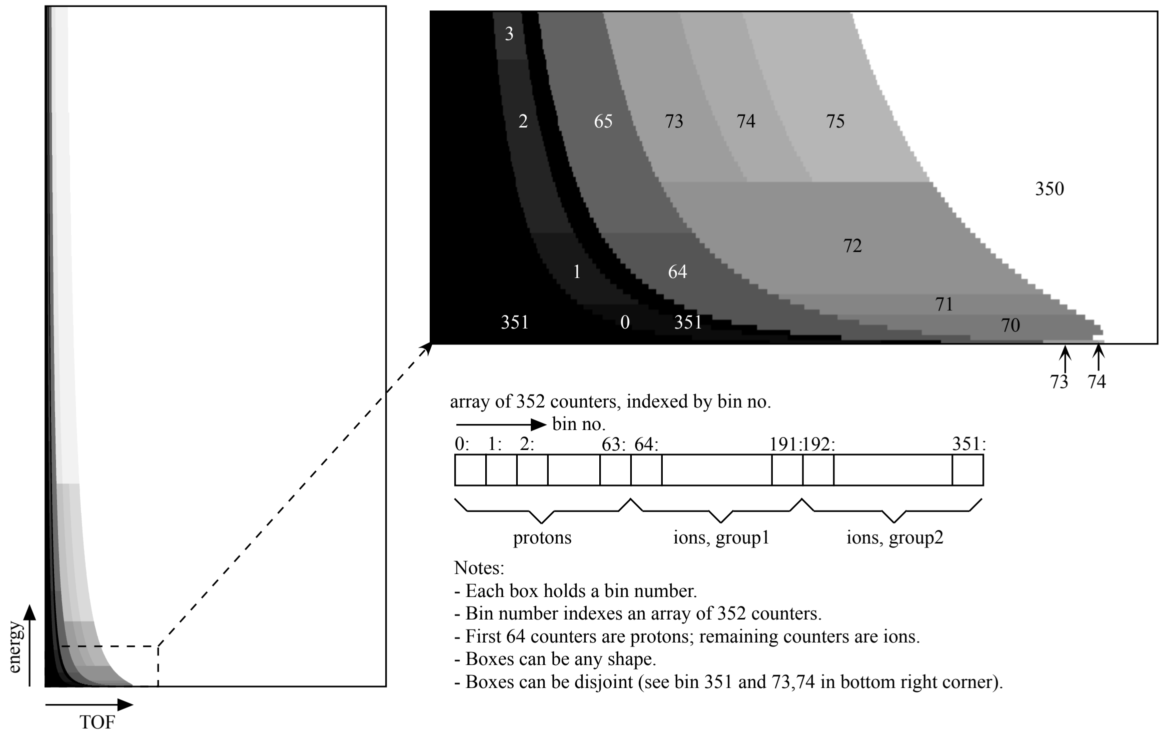
```
                PHThresh[quad, 2]++
            -- compute mass histograms
            TBD: mass histogram here
        else                    -- i.e. TDC-only
            bin = IonTOFBin[ionTOF]
            IonTOF[look, bin]++
    else if pulser pos          -- i.e. pos=20
        IonCompBasic[#ion-comp-basic - 2]++
        bin = 0
    else                        -- i.e. pos unknown
        bin = 0
    if room for raw event
        save raw ion comp (bin)
    IonCompBasic[#ion-comp-basic - 1]++
TBD: where do priority events go?
TBD: where does burst data go?
```

If the position is valid, the look direction (i.e., one of eighty) is computed from the event's quadrant and the position. The ion's TOF is normalized, first per quadrant, then for the path length through the wedge; these corrections are made via lookup tables. If the SSD is valid, i.e. an energy measurement has been made, then ions are categorized into different species and counted. This is done by solving $E=(1/2)*M*V^2$ for M (species), given E (energy) and V (d/TOF). Rather than a calculation, a lookup table indexed by energy and TOF is used. The SSD provides a direct measurement of the energy. To cover higher energies, a so-called extended composite energy is calculated using the SSD energy, SSD energy pulse width, and the extended energy pulse width. The ion's energy is the extended composite energy, normalized per octant. The extended composite energy is calculated as shown in the following pseudo-code.

```
extCompEgy(egy, PW, extPW):
    if egy < compegythresh#
        energy = egy
    else if compegythresh# + PW < extcompegythresh#
        energy = compegythresh# + PW
    else
        energy = extcompegythresh# + extPW
    return energy
```

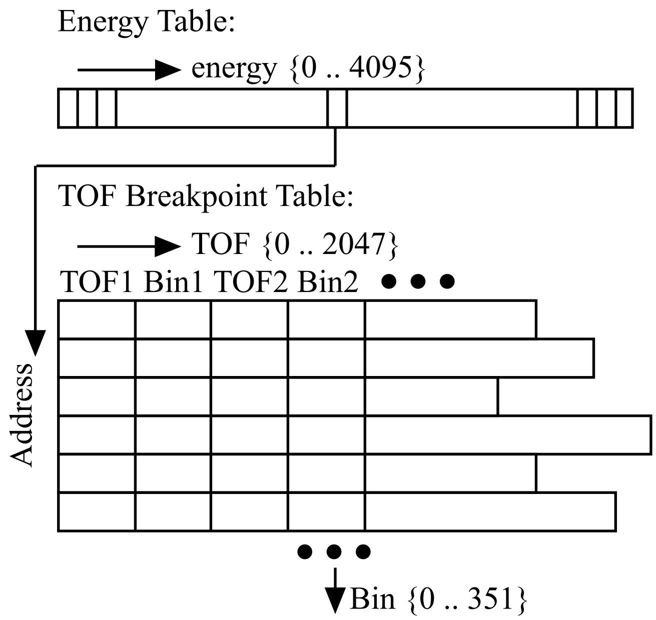
TOF x energy ion species are determined using a lookup table. An example lookup table is shown in the following figure. Different curves correspond to different species. Given an energy and TOF value, the table yields a bin number. The bin number indexes an array of counters. The data processing software does not actually "know" which counters correspond to which species. However, the first 64 counters are for protons. The remaining 288 counters are for ions; these are divided into 128 group1 ions and 160 group2 ions.

Figure 14. Species Determination Lookup Table



A full TOF x energy lookup table would be too large to be practical. Instead an array of arrays implementation is used as shown in the following figure. A table, indexed by the energy, yields a pointer to a TOF breakpoint table. Each TOF breakpoint table represents a slice through the full TOF x energy lookup table. A TOF breakpoint table consists of a series of TOF values and bin numbers. The table is searched with TOF to find the appropriate bin number. Multiple energy table entries can point to the same TOF breakpoint table. This, in effect, computes the logarithm of the energy and commensurately reduces the total table size. The bin number found in the TOF breakpoint table and the event's channel number are used to select a rate bin to increment.

Figure 15. Array of Arrays Species Determination



Each TOF breakpoint table represents a slice through the full TOF x energy lookup table. A TOF breakpoint table consists of a series of TOF values and bin numbers. The TOF values and the bin numbers are 16 bits. The table is a right-to-left slice, i.e. the TOF values must be monotonically decreasing. Each slice can have a different number of TOF breakpoints; however, the last TOF value serves as a sentinel and must be 0. So, Bin1 will be used when TOF1 <= TOF, Bin2 will be used when TOF2 <= TOF < TOF1, Bin3 will be used when TOF3 <= TOF < TOF2, etc. Once the bin number is determined, a series of range tests is performed to determine whether the event is a proton, a group1 ion, or a group 2 ion. The corresponding histogram, indexed by the look direction and bin number, is incremented.

TBD: PH scan and mass histograms

If the event's position is valid, but the SSD is not valid, then it is a TDC-only event. A bin number is found from the ion's normalized TOF value via a lookup table. An ion TOF histogram, indexed by the look direction and bin number, is incremented.

3.3.3 Single Event Data TBD Puck

EPI-Lo saves details of selected ion species events. A priority number from 0 to 5 is assigned to each event; the priority is computed via a lookup table indexed by the event's computed bin number. There are separate lookup tables for TOFxEnergy vs. TOFxPH events, but the priority space spans both event types. The following figure shows an example prioritization. TOFxEnergy events will have priorities 0 through 2; TOFxPH events will have priorities 2 through 5. Note that priority 2 events can be

TOFxEnergy or TOFxPH events in this example. Also note that every possible event will have a priority, i.e. there is a priority defined for every value of TOF, energy, and PH.

Figure 16. Event Prioritization

TBD

The EPI-Lo software can be commanded to only save events of a given priority (EPILO_DAT_PRI_SEL); the software can also be commanded to automatically rotate through all priorities. The events are saved in the order in which they are received, i.e. FIFO order. The maximum number of events to collect is commandable (EPILO_DAT_PRI_EVT); setting the number of events to zero disables the priority event data. The priority event data is integrated for S sectors. The event's TOF, Start Chan, Stop Chan, and SSD Chan are saved. For TOFxEnergy events, the composite energy is saved; for TOFxPH events the MCP PH is saved.

Note: Entries in the priority lookup table can be set to values outside the 0 to 5 priority range. An event that maps to an unknown priority value will be discarded. This could be used to eliminate the most common events. Also, prioritization can be turned off by setting the lookup tables to map to a single priority and selecting that priority.

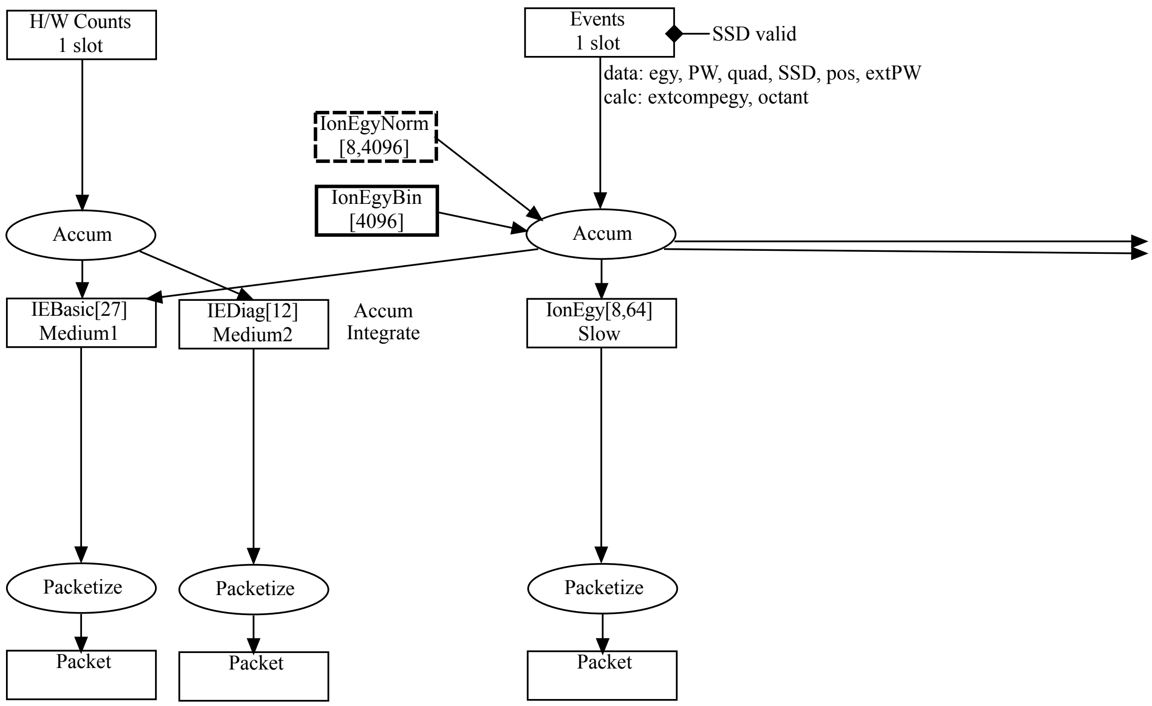
EPI-Lo also collects raw event data. Raw data is collected for each of the three hardware modes: electron energy, ion energy, and ion species. The three types of events are placed in three distinct data products. The events are saved in the order in which they are received, i.e. FIFO order. The maximum number of events to collect is commandable per mode (EPILO_DAT_EVT); setting the number of events to zero disables the product. The raw data is integrated for S sectors.

For ion species events, all of the event data is saved (see Table 2) as well as the computed bin number. For electron energy and ion energy events, the SSD energy, SSD PW, SSD Coincidence, SSD Flags, and SSD Chan are saved. In addition, the computed high energy-resolution bin number is saved.

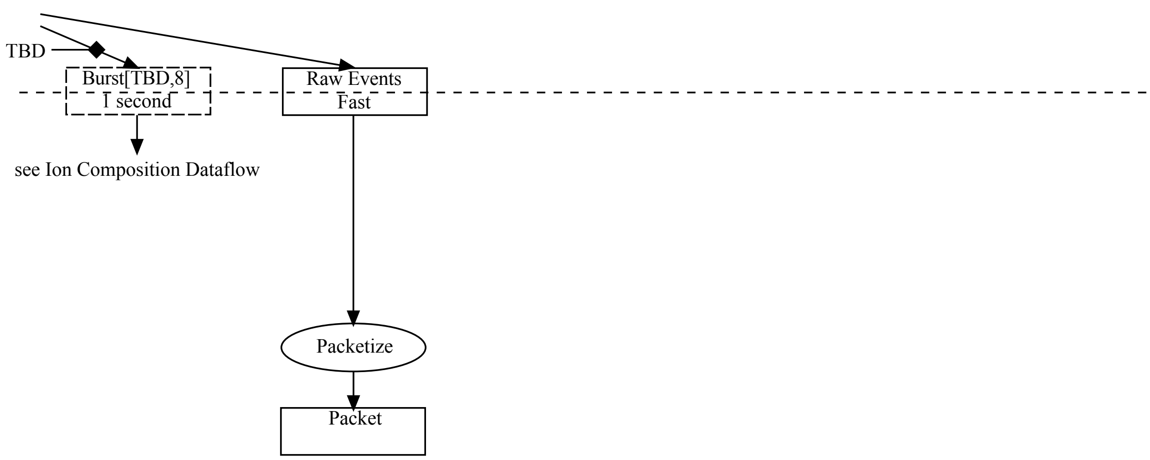
3.4 Ion Energy

EPI-Lo produces ion energy spectra in ion energy mode. The following two figures summarize the data flow from hardware counts and events to telemetry packets. The packets can be categorized into rates, binned data, and single event data; these are described below.

Figure 17. Ion Energy Dataflow: Part 1



Ion Energy Dataflow: Part 2



3.4.1 Rate Counters

The EPI-Lo software integrates the hardware event counters into a basic and diagnostic rates data products. The 21-bit hardware counters are read once per slot and accumulated in 32-bit counters in software. The software also counts the number of pulses it determines are from the internal pulsers and the total number of events it is able to process. The key hardware counters and both software counters comprise the basic rates. Additional hardware counters comprise the diagnostic rates. These are shown in the following two tables.

Table 8. Ion Energy Basic Counters

Name	Description
A Start1 CFD	A CFD pulses
A Stop CFD	
A1 SSD Energy LED	A energy pulses
A2 SSD Energy LED	
A Valid Coincidence	A valid coincidence events
A Event Active Time	A event analysis activity time
...	B pulses
...	C pulses
...	D pulses
FIFO Write	FIFO writes
Internal Pulses	Internal pulse events
Processed	Events processed by software

Table 9. Ion Energy Diagnostic Counters

Name	Description
A Start2 CFD	A CFD pulses
A1 Extended Energy	A energy pulses
A2 Extended Energy	
...	B pulses
...	C pulses
...	D pulses

3.4.2 Binned Event Data

The EPI-Lo software analyzes ion energy event data. Each event read from the FIFO includes a position. If the position indicates that the event was not a test pulse, the ion's energy is computed. The ion's energy is the extended composite energy, normalized per octant; this is the same computation used in ion composition mode with the same normalization lookup table. A bin number is found from the ion's normalized energy via a lookup table. An ion energy histogram, indexed by the octant and bin number, is incremented. If the position indicates that the event was generated by EPI-Lo itself, i.e., a test pulse event, the event is counted in the rates data. This excludes test pulses from the binned event data. Regardless of the position, the event is a candidate for being saved as raw data and will be counted as a processed event in the rates data. This is summarized in the following pseudo-code.

```
accum-ion-energy-event:
    -- Precondition: event ready: egy, PW, quad, SSD, pos extPW
    -- Precondition: SSD valid
    if not pulser pos       -- i.e. pos<>20
        octant = 2*quad + SSD
        ionEgy = IonEgyNorm[octant, extCompEgy(egy, PW, extPW)]
        bin = IonEgyBin[ionEgy]
        IonEgy[octant, bin]++
    else                    -- i.e. pos=20
        IonEgyBasic[#ion-egy-basic - 2]++
        bin = 0
    if room for raw event
        save raw ion energy(bin)
    IonEgyBasic[#ion-egy-basic - 1]++
```

3.4.3 Single Event Data

TBD

3.5 Particle Composition

EPI-Lo produces composition of moderate energy ions and TOF histograms in particle composition mode. The following two figures summarize the data flow from hardware counts and events to telemetry packets. The packets can be categorized into rates, binned data, and single event data; these are described below.

Figure 18. Particle Composition Dataflow: Part 1

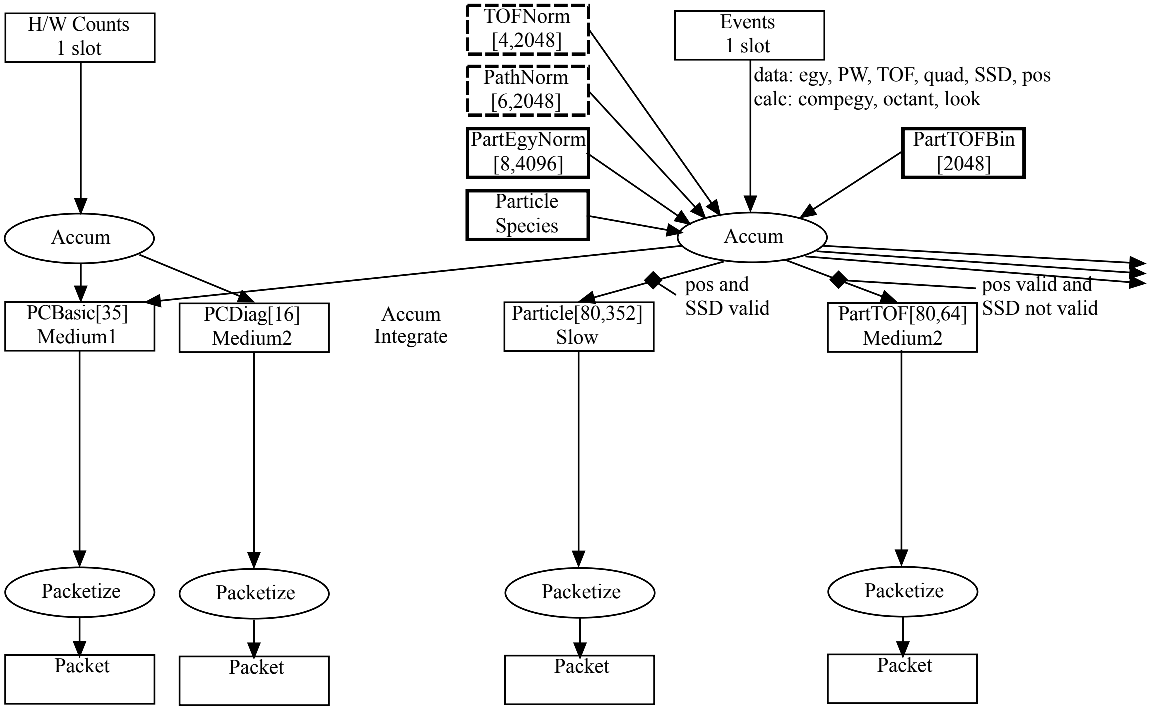
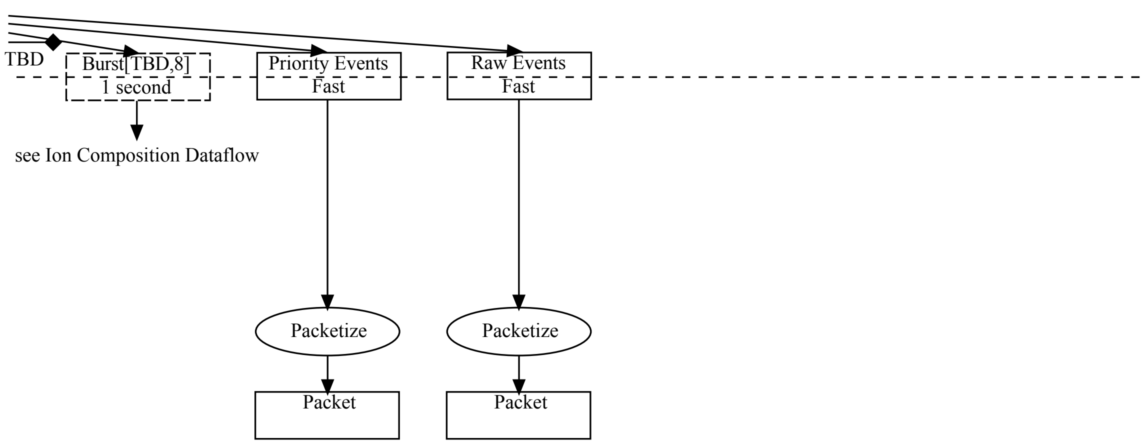


Figure 19. Particle Composition Dataflow: Part 2



3.5.1 Rate Counters

The EPI-Lo software integrates the hardware event counters into a basic and diagnostic rates data products. The 21-bit hardware counters are read once per slot and accumulated in 32-bit counters in software. The software also counts the number of pulses it determines are from the internal pulsers and the total number of events it is able to process. The key hardware counters and both software counters comprise the basic rates. Additional hardware counters comprise the diagnostic rates. These are shown in the following two tables.

Table 10. Particle Composition Basic Counters

Name	Description
A Start1 CFD	A CFD pulses
A Stop CFD	
A1 SSD Energy LED	A energy pulses
A1 Extended Energy	
A2 SSD Energy LED	
A2 Extended Energy	
A Valid Coincidence	A valid coincidence events
A Event Active Time	A event analysis activity time
...	B pulses
...	C pulses
...	D pulses
FIFO Write	FIFO writes
Internal Pulses	Internal pulse events
Processed	Events processed by software

Table 11. Particle Composition Diagnostic Counters

Name	Description
A Start2 CFD	A CFD pulses
A Start1 to Stop TDC VE	A TOF chip VE pulses
A Start2 to Stop TDC VE	
A Start1 to Start2 TDC VE	
...	B pulses
...	C pulses
...	D pulses

3.5.2 Binned Event Data

The EPI-Lo software analyzes particle composition event data. The analysis is very similar to that performed on ion composition events. Each event read from the FIFO includes a position. If this is one of the twenty valid positions, the event will be binned as described below. If the position indicates that the event was generated by EPI-Lo itself, i.e., a test pulse event, the event is counted in the rates data. This excludes test pulses from the binned event data. Regardless of the position, the event is a candidate for being saved as raw data and will be counted as a processed event in the rates data. This is summarized in the following pseudo-code.

```
accum-part-comp-event:
    -- Precondition: event ready: egy, PW, TOF, SSD, pos
    if pos valid            -- i.e. pos {0..19}
        look = 20*quad + pos
        partTOF = PathNorm[Path[pos], TOFNorm[quad, TOF]]
        if SSD valid        -- i.e. not TDC-only
            octant = 2*quad + SSD
            partEgy = PartEgyNorm[octant, compEgy(egy, PW)]
            bin = PartSpecies[partEgy, partTOF]
            Particle[look, bin]++
        else                -- i.e. TDC-only
            bin = PartTOFBin[partTOF]
            PartTOF[look, bin]++
    else if pulser pos      -- i.e. pos=20
        PartCompBasic[#part-comp-basic - 2]++
        bin = 0
    else                    -- i.e. pos unknown
        bin = 0
    if room for raw event
        save raw particle comp (bin)
    PartCompBasic[#part-comp-basic - 1]++
TBD: where do priority events go?
```

If the position is valid, the look direction (i.e., one of eighty) is computed from the event's quadrant and the position. The particle's TOF is normalized, first per quadrant, then for the path length through the wedge. These corrections are made via lookup tables; these are the same lookup tables used to normalize the TOF of ion composition events. If the SSD is valid, i.e. an energy measurement has been made, then particles are categorized into different species and counted. This is done by solving $E=(1/2)*M*V^2$ for M (species), given E (energy) and V (d/TOF). Rather than a calculation, a lookup table indexed by energy and TOF is used. The particle's energy is the composite energy, normalized per octant. Similar to the extended composite energy, the composite energy is calculated as shown in the following pseudo-code.

```
compEgy(egy, PW):
    if egy < compegythresh#
        energy = egy
    else
        energy = compegythresh# + PW
    return energy
```

TOF x energy particle species are determined using a lookup table. The same array of arrays species determination method used to identify ion composition species is used (see Figure 14 and Figure 15). Once the bin number is determined, a particle histogram, indexed by the look direction and bin number, is incremented.

If the event's position is valid, but the SSD is not valid, then it is a TDC-only event. A bin number is found from the particle's normalized TOF value via a lookup table. A particle TOF histogram, indexed by the look direction and bin number, is incremented.

3.5.3 Single Event Data

TBD

3.6 Particle Energy

EPI-Lo produces electron energy spectra and composition of high energy ions in particle energy mode. The following two figures summarize the data flow from hardware counts and events to telemetry packets. The packets can be categorized into rates, binned data, and single event data; these are described below.

Figure 20. Particle Energy Dataflow: Part 1

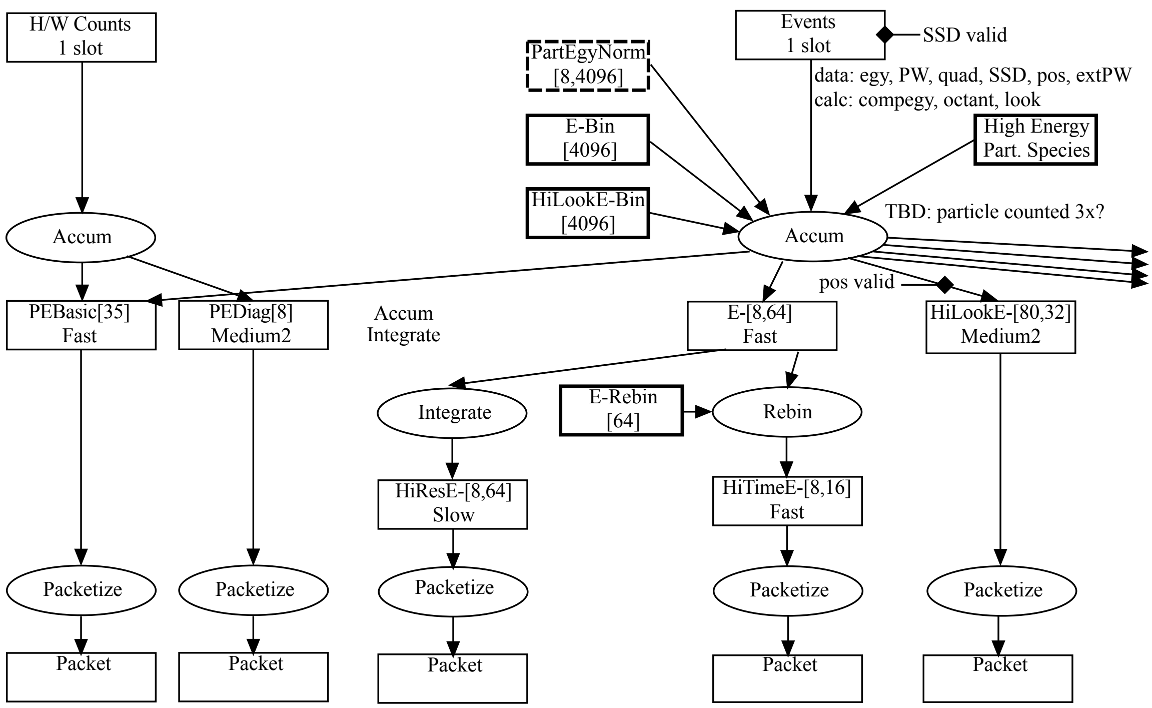
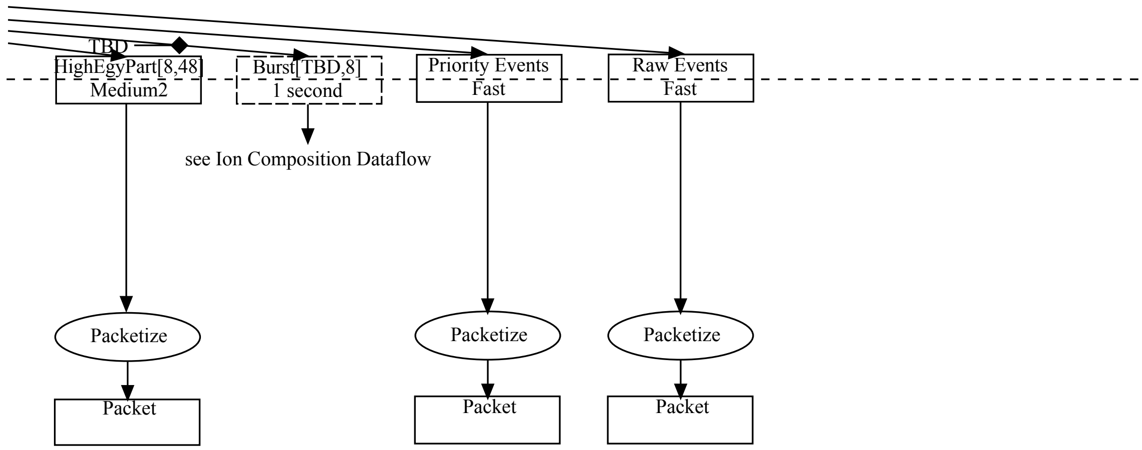


Figure 21. Particle Energy Dataflow: Part 2



3.6.1 Rate Counters

The EPI-Lo software integrates the hardware event counters into a basic and diagnostic rates data products. The 21-bit hardware counters are read once per slot and accumulated in 32-bit counters in software. The software also counts the number of pulses it determines are from the internal pulsers and the total number of events it is able to process. The key hardware counters and both software counters comprise the basic rates. Additional hardware counters comprise the diagnostic rates. These are shown in the following two tables.

Table 12. Particle Energy Basic Counters

Name	Description
A Start1 CFD	A CFD pulses
A Stop CFD	
A1 SSD Energy LED	A energy pulses
A1 Extended Energy	
A2 SSD Energy LED	
A2 Extended Energy	
A Valid Coincidence	A valid coincidence events
A Event Active Time	A event analysis activity time
...	B pulses
...	C pulses
...	D pulses
FIFO Write	FIFO writes
Internal Pulses	Internal pulse events
Processed	Events processed by software

Table 13. Particle Energy Diagnostic Counters

Name	Description
A Start2 CFD	A CFD pulses
A Start1 to Start2 TDC VE	A TOF chip VE pulses
...	B pulses
...	C pulses
...	D pulses

3.6.2 Binned Event Data

The EPI-Lo software analyzes particle energy event data. Each event read from the FIFO includes a position. If the position indicates that the event was not a test pulse, the event will be binned as described below. If the position indicates that the event was generated by EPI-Lo itself, i.e., a test pulse event, the event is counted in the rates data. This excludes test pulses from the binned event data. Regardless of the position, the event is a candidate for being saved as raw data and will be counted as a processed event in the rates data. This is summarized in the following pseudo-code.

```
accum-part-energy-event:
    -- Precondition: event ready: egy, PW, SSD, pos, extPW
    -- Precondition: SSD valid
    if not pulser pos       -- i.e. pos<>20
        octant = 2*quad + SSD
        -- bin electrons
        partEgy = PartEgyNorm[octant, compEgy(egy, PW)]
        bin = E-Bin[partEgy]
        E-[octant, bin]++
        if pos valid        -- i.e. pos {0..19}
            -- if known position, bin high-look electrons
            look = 20*quad + pos
            bin2 = HiLookE-Bin[partEgy]
            HiLookE-[look, bin2]++
        -- bin high energy particles
        bin3 = HighEgyPartSpecies[partEgy, extPW] TBD: need extPW
normalization?
        HighEgyPart[octant, bin3]++
    else                    -- i.e. pos=20
        PartEgyBasic[#part-egy-basic - 2]++
        bin = 0
    if room for raw event
        save raw particle energy (bin)
    PartEgyBasic[#part-egy-basic - 1]++
TBD: where do priority events go?
```

If the position indicates that the event was not a test pulse, the particle's energy is computed. The particle's energy is the composite energy, normalized per octant; this is the same computation used in particle composition mode with the same normalization lookup table. A bin number is found from the particle's normalized energy via a lookup table. An electron energy histogram, indexed by the octant and bin number, is incremented.

A small percentage of the particles will have an accompanying position measurement. If the position is valid, the look direction (i.e., one of eighty) is computed from the event's quadrant and the position. A bin number is found from the particle's normalized energy, computed above. via a lookup table. A high-look resolution electron energy histogram, indexed by the position and bin number, is incremented.

The composition of high-energy ions using the particle energy, computed above, and the pulse width from the anti-coincidence detector. Species composition is determined using a lookup table. The same array of arrays species determination method used to identify ion composition species is used (see Figure 14 and Figure 15). Once the bin number is determined, a high-energy particle histogram, indexed by the octant and bin number, is incremented.

3.6.3 Single Event Data

TBD

3.7 Lookup Tables TBD Puck

The EPI-Lo lookup tables are summarized below. There are energy binning tables as well as rebinning tables for both ion and electron energies. There are also TOF x energy and TOF x pulse height species determination tables and proton rate rebinning tables.

Table 14. EPI-Lo Lookup Tables

MRAM Address	Name	Length (bits)	Value	Description
0x1a0000	Pulse Width Correction	6 * 16	Unsigned integer	Pulse width correction table (scaled by 10,000)
0x1a000c	Electron Energy Bin	2048 * 8	0 – 63	Electron energy binning table
0x1a080c	Ion Energy Bin	2048 * 8	0 – 63	Ion energy binning table
0x1a100c	TOF x Energy: Energy to Index	2048 * 16	0 - 16383	TOF x energy species determination table. First table maps energy into an index into second table, the TOF breakpoint table.
0x1a200c	TOF x Energy: TOF Breakpoints	16384 * 8	See Figure 15	
0x1a600c	TOF x Energy: Event Priority	34 * 8	0 – 5	TOF x energy event priority
0x1a602e	TOF x PH: Energy to Index	2048 * 16	0 - 16383	TOF x pulse height species determination table. First table maps energy (pulse height) into an index into second table, the TOF breakpoint table.
0x1a702e	TOF x PH: TOF Breakpoints	16384 * 8	See Figure 15	
0x1ab02e	TOF x PH: Event Priority	32 * 8	0 – 5	TOF x PH event priority
0x1ab04e	Electron Energy Rebin	64 * 8	0 – 13	Electron energy rebinning table, maps high-resolution to low-resolution bins
0x1ab08e	Space Weather Proton Bins	4 * 8	0 – 13	Space weather parameters: proton bin numbers, energy bin numbers, and start and SSD channels for each
0x1ab092	Space Weather Energy Bins	2 * 8	0 – 63	
0x1ab094	Space Weather Proton Start	8	0 - 5	
0x1ab095	Space Weather Energy SSD	8	0 - 5	
0x1ab099	TOF x PH Proton Rebin	32 * 8	0 – 9	TOF x pulse height proton rebinning table maps high-resolution to low-resolution bins
0x1ab0b6	*Reserved*	20298 * 8		

The tables are stored in MRAM; the address of each table is listed above. On start-up, the tables are copied into RAM. A reload of the tables into RAM can be forced by command (EPILO_DAT_RECALC). This might be used if the tables in MRAM have changed or parameters affecting the tables have been modified. Note: an area at the end of the MRAM is reserved for the user, e.g. for descriptive text; this area is not used by the software.

4. Other Subsystems

Other major EPI-Lo subsystems include the power supplies and the internal pulser. The operation of these subsystems is discussed below.

4.1 SSD Bias and MCP High Voltage Power Supplies

EPI-Lo has two power supplies, one for the SSD bias voltage and one for the MCP high voltages. It takes at least two commands to turn on either supply. Before setting any high voltage (HV), the supply must be enabled by command (EPILO_BV_ENB and EPILO_HV_ENB). Enabling a supply turns on the supply's clock. Disabling the supply turns off its clock; all control voltage DACs will be set to zero.

Each supply has an upper limit and a goal; each of these must be commanded to turn on the supply. For the SSD bias voltage, the limit is commanded (EPILO_BV_LIMIT), and then the goal is commanded (EPILO_BV_LEVEL). The commanded goal must be less than or equal to the commanded limit. The goal may also be stepped by command (EPILO_BV_STEP). The EPI-Lo software automatically ramps the voltage to its commanded goal; a DAC is written with the control voltage every second. If a new goal is set before the old goal is reached, the new goal is followed. The ramp-up rate and ramp-down rate are uploadable parameters; the ramp-up rate is in seconds per step and the ramp-down rate is in steps per second.

Operating the MCP high voltages is more complex. There is a common bulk high voltage and four individual MCP high voltages derived from the common bulk high voltage. The four MCP HVs supply the MCPs in the four sensor quadrants. The MCP HVs must always be set to a value less than the common bulk HV; this is enforced by the EPI-Lo software. The common bulk HV limit and goal are set by command (EPILO_HV_COM_LIMIT and EPILO_HV_COM_LEVEL). The commanded goal must be less than or equal to the commanded limit. Further, the commanded goal must be greater than or equal to the commanded goal of every MCP HV. The limits and goals of the individual MCP HVs are also set by command (EPILO_HV_MCP_LIMIT and EPILO_HV_MCP_LEVEL). The commanded goal must be less than or equal to the commanded limit. Further, the commanded goal must be less than or equal to the common bulk HV goal. The MCP HV goals may also be stepped by command (EPILO_HV_MCP_STEP). The EPI-Lo software automatically ramps each voltage to its commanded goal. DACS are written with the control voltages every second; in the case of the common bulk HV DAC, an offset of TBD is added. If a new goal is set before the old goal is reached, the new goal is followed. The ramp-up rate and ramp-down rate are uploadable parameters (distinct from the SSD bias voltage rates); the ramp-up rate is in seconds per step and the ramp-down rate is in steps per second.

Each MCP HV supply has a hardware current limit: if the HV's current exceeds the limit, the supply is automatically shut down via a fast hardware response TBD: elaborate on this. The automatic shutdown can be enabled or disabled by command (EPILO_MCP_CUR_ENB). The current limit is also commandable (EPILO_MCP_CUR_LIM). The software detects that the hardware has shut down the HV and issues an alarm, zeroes the HV's control DAC, and runs a macro. The HV will remain shut down until the current comes back within limits and the automatic shutdown is disabled.

The common bulk HV supply is also controlled by an external safing plug. If the plug is installed, the common bulk HV can be turned on, but the output voltage is limited in hardware such that the HV can be operated in air. The EPI-Lo software also monitors the safing plug, and clamps the maximum common bulk HV limit that can be commanded (EPILO_HV_COM_LIMIT) and thus reduces the maximum common bulk HV goal that can be commanded (EPILO_HV_COM_LEVEL) and the maximum MCP HV goal that can be commanded (EPILO_HV_MCP_LEVEL). The limit is an uploadable parameter.

4.2 Diagnostic and Test Support

Pulses can be injected into the front-end circuitry for test purposes. A start pulse and a stop pulse, as well as two SSD pulses, are generated. The rate of the pulses, the delay from the start pulse to the stop pulse, the delay from the start pulse to the SSD1 pulse, and the delay from the start pulse to the SSD2 pulse can be commanded (EPILO_TST_PUL_CFG). The start and stop pulses for the individual quadrants can be enabled or disabled by command (EPILO_TST_PUL_ENB). All of the SSD1s and all of the SSD2s are enabled or disabled as a group (also via EPILO_TST_PUL_ENB). The amplitude of the start/stop pulse and the SSD pulse can be set by command (EPILO_TST_PUL_HGT).

The EPI-Lo hardware can be commanded (EPILO_TST_BASE) to measure SSD energy channel baseline values instead of doing its normal event processing. The results appear in the event FIFO. Most of the event data is zeros as shown in the following table. SSD Energy contains the ADC value direct from the electronics, i.e., without the current baseline removed. SSD Chan and Quadrant indicate the channel being measured. The Position will indicate "invalid".

Table 15. Event Data (In Baseline Mode)

Name	Bits	Description

Start1 to Stop TDC	11	N/A (Zeros)
Start1 to Start2 TDC	11	N/A (Zeros)
Start2 to Stop TDC	11	N/A (Zeros)
Start2 to LED TDC	11	N/A (Zeros)
TDC Valid Flags	4	N/A (Zeros)
MCP PH Flags	2	N/A (Zeros)
TOF	11	N/A (Zeros)
Position	5	Calculated position 31 = invalid
SSD Energy	12	SSD energy
SSD Energy PW	12	N/A (Zeros)
SSD Ext. Energy PW	12	N/A (Zeros)
SSD Coin Flags	5	N/A (Zeros)
SSD Chan	2	SSD channel 0 = SSD1, 1 = SSD2, 2-3 = invalid
Quadrant	2	Quadrant 0 = A, 1 = B, 2 = C, 3 = D

EPI-Lo can also export internal signals to a test port. Up to five different signals can be monitored simultaneously. The signals to export are selectable by command (EPILO_TST_SIGNAL). This is only useful during ground testing.

5. Support Processing

5.1 Alarms and Monitoring

Alarms report problems found by the software. Each alarm is described by an ID, two values, and a flag. The ID identifies the problem that has occurred and the accompanying values offer additional information. The flag indicates whether the alarm was caused by a transient or a persistent condition. See Table 72 for a list of alarms.

The alarms are divided into two groups: one for reporting internal software problems and another for reporting out-of-limit conditions for monitored data. Software problems are all reported as transient alarms. When the problem occurs, the alarm is generated and the software recovers from the problem as best it can.

A collection of environmental data is monitored by EPI-Lo. There are analogs including voltages, currents, and temperatures read and monitored via the processor's SPI bus. The analog values are read every second. The monitoring is done on every third sample, i.e. every three seconds.

Each monitored item has a lower and upper limit. If an item is out of limits one cycle, but back within limits on the subsequent cycle, a transient alarm is reported. The alarm ID indicates the item being monitored and whether the value was too low or too high. The values accompanying the alarm are the out of limits data and the corresponding limit. If an item is either too high or two low for two consecutive monitoring cycles, a persistent alarm is reported. Again, the ID indicates the item being monitored, etc. The values accompanying the alarm are the second out-of-limits data value and the limit. If enabled via the EPILO_MON_CNTRL command, the software will also act to eliminate the problem; a high or low response macro designated for the alarm is run.

If the item is out of limits for more than two cycles, indicating that the first response failed to eliminate the problem, more drastic action is taken. Monitors are divided into three classes; the class determines what action is taken. For no-operation (nop) monitors, nothing is done. This class is most frequently used for temperature monitors. For retry monitors, if enabled, the response macro is rerun. This is most useful for count rate monitors. For shutdown monitors, if enabled, the shutdown macro is run. This class is most useful for current or voltage monitors.

The following pseudo-code description of the reactions to a high monitor summarizes the discussion above. The low responses are similar.

```
high once:
      issue transient high alarm

high twice:
      issue persistent high alarm
      if enabled (via EPILO_MON_CNTRL command)
            run high response macro for this alarm

high more than twice:
      case of monitor class
            nop:
                  do nothing
            retry:
                  if enabled (via EPILO_MON_CNTRL command)
                        run high response macro for alarm
            shutdown:
                  if enabled (via EPILO_MON_CNTRL command)
                        run shutdown macro
```

The monitored data is summarized in the following table. The monitor class is encoded as S=shutdown, N=nop, and R=redo. The reported alarm Ids are for low and high excursions; similarly there are low and high response macro Ids. An empty macro id field denotes macro 0, the “No action” macro.

Table 16. Monitored Data TBD Puck

Source	Class	Alarm Ids Low / High		Macro Ids Low / High	
+1.5V voltage	S	128	192	4	4
+3.3V voltage	S	129	193	4	4
+5V voltage	S	130	194	4	4
+15V voltage	S	131	195	5	4
+15V current	S	132	196		5
+5V current	S	133	197		4
+3.3V current	S	134	198		4
+1.5V current	S	135	199		4
CFD voltage	N	136	200		
Primary current	S	137	201		4
SSD bias voltage power	N	138	202		6
SSD bias voltage	S	139	203		6
MCP HV current	S	140	204		5
MCP HV voltage	S	141	205		5
LVPS temperature	N	142	206		1
Puck temperature	N	143	207		1

5.2 Fault Avoidance and Recovery

Many of the processor's hardware control registers contain triple redundancy and voting logic for each bit. The software periodically rewrites these registers to correct single bit errors.

The processor has a watchdog timer. If the watchdog timer is not tickled from time to time, the processor is reset. The watchdog timeout is 4.096 seconds. The watchdog does not run until it has been enabled; once enabled, it can never by disabled except by processor reset, watchdog or otherwise. One of the EPI-Lo software processes tickles the watchdog once per second. If any process crashes, the process that tickles the watchdog will not get a chance to run and thus the watchdog will timeout and reset the processor.

If EPI-Lo fails to receive any good communications from the spacecraft for some time interval, it will automatically run its safing macro. Once good communications resume, the monitoring is continued so that subsequent loss of communications will trigger safing again. The timeout interval is set by command (EPILO_SAF_TIMEOUT).

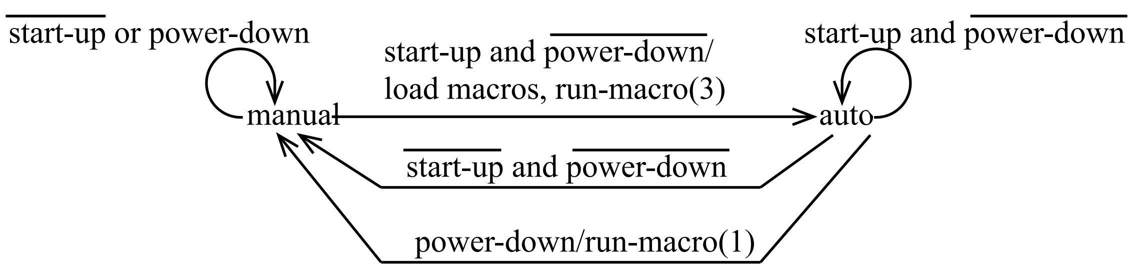
The EPI-Lo application can be commanded to return to boot (EPILO_SAF_RESET). This is implemented by allowing the watchdog timer to go off and thus produces a hardware reset. The EPI-Lo application can request that the spacecraft turn off EPI-Lo's power (EPILO_SAF_OFF). The EPI-Lo application can also request that the spacecraft cycle EPI-Lo's power, i.e. off, then back on (EPILO_SAF_CYCLE).

5.3 Autonomous Operations

EPI-Lo must be able to operate autonomously, i.e. independently of ground or spacecraft commands. So, from power-up, EPI-Lo must be able to boot, configure its subsystems, including turning on high-voltages, and start collecting science data. The boot program examines a start-up flag in the time and status message received from the spacecraft. If the flag indicates that autonomous operation is required, then the boot program starts the EPI-Lo application program stored in non-volatile memory (see 0).

The EPI-Lo application program also examines the start-up flag. If the flag indicates autonomous operations, macros stored in non-volatile memory are loaded. Then, one macro, designated the start-up macro, is executed. This prepares the EPI-Lo instrument for science operations, e.g. by ramping up the high voltages. Once in autonomous operations mode, or "auto" mode, the EPI-Lo software continues to monitor the start-up flag; it also monitors a power-down warning flag in the time and status. If the power-down flag is set, EPI-Lo runs its shutdown macro and leaves auto mode. If the start-up flag is reset, EPI-Lo also leaves auto mode; this should not happen in normal operations but is useful in testing. These operations are summarized in the following figure.

Figure 22. Autonomous Operations Modes

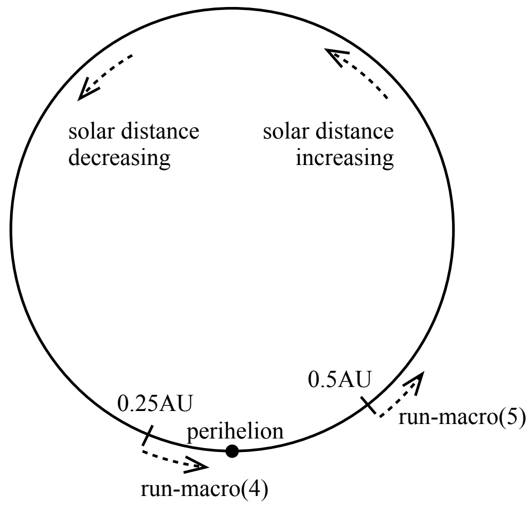


Note: if loading macros from non-volatile memory fails, i.e. because of a bad checksum, an alarm is generated and the macros currently in memory will be used instead.

While in auto mode, the EPI-Lo software will monitor the solar distance, again available in the time and status message. Comparing the solar distance against different thresholds, the application program will select a science macro to run. This will select the science products, integration cadence, etc. appropriate for EPI-Lo's position in the orbit. As different thresholds are crossed, different macros are run, reconfiguring EPI-Lo appropriately. Each threshold and macro define an autonomous operations region. Up to 25 regions can be defined by command (EPILO_OP_REGION). The following figure shows an example with two operations regions defined. One region is between 0.25AU towards perihelion to 0.5AU away from perihelion. When this region is entered, macro 4 is run. The second region covers everywhere in the orbit not covered by the first region. When this region is entered, i.e. at 0.5AU going away from perihelion, macro 5 is run.

Figure 23. Operations Regions Definition Example

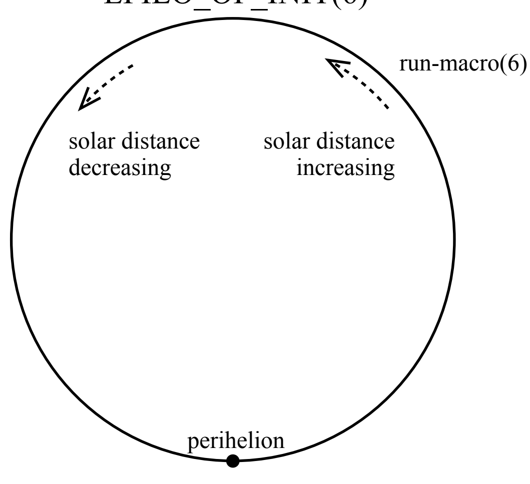
EPILO_OP_REGION(4, Decreasing, 0.25AU)
EPILO_OP_REGION(5, Increasing, 0.5AU)



All defined regions can be deleted by command (EPILO_OP_INIT). The command includes a macro to run. This macro will be run when auto mode is entered; or, if already in auto mode, the macro will be run shortly after the command.

Figure 24. Operations Regions Initialization Example

EPILO_OP_INIT(6)



5.4 Start-Up

On start-up, the MCP HV and SSD bias voltage are turned off (this is actually done in hardware). All subsystems, including the MCP HVs and SSD bias voltage are disabled. All data products are disabled. Finally, all of the discriminator thresholds are set to their maximum values. TBD

6. Host I/O

EPI-Lo communicates with its host, the SPP spacecraft, via a Universal Asynchronous Receiver Transmitter (UART) serial interface (see Reference 1). The interface operates at 115200 baud with eight data bits, odd parity, and one stop bit. Data is sent over the command and telemetry interfaces as a transaction. For all multi-bit or multi-byte numbers, the first byte sent has the most numerical significance; this is so-called "big-endian" format. This convention applies to arguments in commands and data in telemetry.

The interface is redundant. The S/C sends commands to EPI-Lo via the active side of the redundant interface. EPI-Lo determines which side is active and sends telemetry on that side of the redundant interface. The active side can change dynamically; EPI-Lo tracks these changes, although some command and telemetry data may be lost.

Each transaction contains a single Instrument Transfer Frame (ITF). Each ITF contains command or telemetry data, in CCSDS format. The format of the ITF is shown in the following table.

Table 17. Instrument Transfer Frame (ITF) Format

Name	Length (bits)	Value	Description
Sync	4 * 8	0xfefa30c8	Synchronization pattern
Virtual Channel	8	0x05	Fixed value for each instrument interface
Sequence	8	Unsigned integer	Sequence number per virtual channel
Length	16	Unsigned integer	Number of bytes following the Length field, including the Checksum
First Header	16	Unsigned integer (0xffff = no header)	Offset of first packet header in the Data (0 for commands)
Data	N * 8		Command or telemetry data
Checksum	16		16-bit exclusive-or of ITF, except for sync

The spacecraft sends one command ITF to EPI-Lo every second. EPI-Lo derives a one-pulse-per-second (1PPS) from the command ITF. This "virtual" 1PPS is signaled by the falling edge of the start bit of the first byte of the ITF. EPI-Lo sends one telemetry ITF to the spacecraft every second; the ITF cannot straddle the 1PPS.

7. Command Interface

EPI-Lo receives and processes command ITFs from the spacecraft. A command ITF can be at most 512 bytes, including all headers. The length of the command ITF must be an even number of bytes. EPI-Lo searches for the synchronization pattern for each frame received. If synchronization is lost, bytes are discarded until synchronization is restored. The length and checksum of every frame are verified. An invalid frame is viewed as a communications problem; the frame is discarded and an alarm is generated.

Table 18. Command ITF Format

Name	Length (bits)	Value	Description
Sync	4 * 8	0xfefa30c8	Synchronization pattern
Virtual Channel	8	0x05	Fixed value for each instrument interface *(Unused)*
Sequence	8	Unsigned integer	Sequence number per virtual channel *(Unused)*
Length	16	4 - 504	Number of bytes following the Length field, including the Checksum
First Header	16	0	Offset of first packet header in the Data *(Unused)*
Data	N * 8		Command data
Checksum	16		16-bit exclusive-or of ITF, except for sync

After the frame is verified, EPI-Lo processes CCSDS packets in order. If a CCSDS header is badly formatted, e.g. it has a bad length, the CCSDS packet and the rest of the frame are discarded and an alarm is generated. If a CCSDS packet primary header has an incorrect version or type, or has an unknown APID, the packet is discarded and an alarm is generated; processing continues through the rest of the frame. Similarly, if a CCSDS packet has an incorrect length for its APID, the packet is discarded, an alarm is generated, and processing resumes.

A frame will contain time and status, followed by zero, one, or more commands, each with CCSDS telecommand headers. The number of commands is limited only by the maximum size of the command ITF. Each of the telecommand types that can be received on the command interface are described below.

7.1 Time and Status

EPI-Lo's internal clock consists of a 32-bit field containing whole seconds of MET and a 16-bit field containing milliseconds within the second. Time and status is delivered from the spacecraft every second; the format is shown in the following table. Note that this is a telemetry packet. The first field following the secondary header contains the Mission Elapsed Time (MET) of the next 1PPS. EPI-Lo sets the whole seconds of its internal clock to this time when the next 1PPS occurs; the milliseconds field is zeroed. The internal time, whole seconds and milliseconds, is updated from a 1kHz source internal to EPI-Lo. If the time and status fails to arrive, the time continues to update using only the internal 1kHz. The 32-bit time is used to tag science and engineering telemetry. The milliseconds-resolution time is used to schedule internal EPI-Lo activities.

Table 19. Time and Status Format

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	0x1df	Application Process ID
Grouping	2	11 = None	Grouping flags *(Unused)*
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID *(Unused)*
Length	16	131	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
MET 1PPS	32	Unsigned integer	Time (MET) of the next 1PPS
Solar Distance	32	Unsigned integer	Current distance to the sun (TBD units)
Solar Array +Y Flap Angle	8	0 - 90	Solar array flag angles (degrees)
Solar Array –Y Flap Angle	8	0 - 90	
Solar Array +Y Feather Angle	8	0 - 90	Solar array feature angles (degrees)
Solar Array –Y Feather Angle	8	0 - 90	
ISIS EPI LO SSR Allocation Status	8	0 - 255	SSR percent full (255 = 100%)
ISIS EPI HI SSR Allocation Status	8	0 - 255	
WISPR SSR Allocation Status	8	0 - 255	
FIELDS 1 (DCB SSR	8	0 - 255	

Name	Length (bits)	Value	Description
Allocation Status			
FIELDS 2 (TDS) SSR Allocation Status	8	0 - 255	
SWEAP SSR Allocation Status	8	0 - 255	
ISIS EPI-LO Shared Data	32	TBD	Shared data from instruments
ISIS EPI-HI Shared Data	32	TBD	
WISPR Shared Data	32	TBD	
FIELDS 1 (DCB) Shared Data	256	TBD	
FIELDS 2 (TDS) Shared Data	256	TBD	
SWEAP Shared Data	240	TBD	
FIELDS 1 (DCB) Rate	1	0 = 80kb 1= 240kb	Data rate control (for FIELDS and SWEAP)
SWEAP Rate	1		
WISPR Rate	1	0 = Off 1 = On	Data rate control for WISPR images
Spare	1	0	
Thruster Fire Status	1	0 = Not firing 1 = Firing	Indicates thruster firing or about to fire
Instrument Shutdown	1	0 = Ok 1 = Shutdown	Indicates imminent instrument shutdown
SBC Transition	1	0 = No transition 1 = Transition	Indicates that the S/C processor has transitioned
Solar Distance	1	0 = Decreasing 1 = Increasing	Indicates whether solar distance is decreasing or increasing

Name	Length (bits)	Value	Description
Inflection			
ISIS EPI LO Shared Data Received	1	0 = Not received 1 = Received	Indicates whether latest shared instrument data has been received and is valid; “not received” indicates that the shared instrument data is old, stale or not valid
ISIS EPI HI Shared Data Received	1	0 = Not received 1 = Received	
WISPR Shared Data Received	1	0 = Not received 1 = Received	
FIELDS 1 (DCB) Shared Data Received	1	0 = Not received 1 = Received	
FIELDS 2 (TDS) Shared Data Received	1	0 = Not received 1 = Received	
SWEAP Shared Data Received	1	0 = Not received 1 = Received	
Solar Distance Valid	1	0 = Not valid 1 = Valid	Indicates whether solar distance is valid
Solar Array +Y Valid	1	0 = Not valid 1 = Valid	Indicates whether solar array flag and feather angles are valid
Solar Array –Y Valid	1	0 = Not valid 1 = Valid	
ISIS EPI LO Startup Mode	1	0 = Manual 1 = Autonomous	Indicates whether instrument should startup manual or autonomous operations mode
ISIS EPI HI Startup Mode	1	0 = Manual 1 = Autonomous	
WISPR Startup Mode	1	0 = Manual 1 = Autonomous	
FIELDS 1 (DCB) Startup Mode	1	0 = Manual 1 = Autonomous	

Name	Length (bits)	Value	Description
FIELDS 2 (TDS) Startup Mode	1	0 = Manual 1 = Autonomous	
SWEAP Startup Mode	1	0 = Manual 1 = Autonomous	
Spare	9	0	

The time and status data also includes solar distance, its inflection and validity flags, the EPI-Lo startup flag, and the instrument shutdown flag. These are used by the EPI-Lo software in autonomous operations; the remaining data are not used.

7.2 Commands

Each command uses CCSDS format. A telecommand packet can be up to 272 bytes, including all headers. The length must be an even number of bytes. In the primary header, Application Process ID (APID) is a constant. The grouping flags in the header will be set to "none"; the grouping flags and the sequence count are not used. The length field counts the number of bytes in the packet after the primary header, including the checksum at the end of the packet. Note that there is no secondary header.

Command messages have an 8-bit opcode that uniquely identifies the command EPI-Lo should execute. This is followed by a macro field that indicates whether the command should be executed immediately or "learned" as part of a macro. Following the macro field are the arguments. Most arguments are a single byte; larger arguments are spread over multiple bytes with the most significant byte first. The last argument may be followed by zero-fill padding to force the size of the command to a multiple of 2 bytes. The format of the command packet is shown below.

Table 20. Command Format

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags *(Unused)*
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID *(Unused)*
Length	16	1 - 265	Packet length (bytes) - 1
Opcode	8		Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Argument 1	8		First argument byte
Argument 2	8		Second argument byte
…			Remaining argument bytes
Pad	0-8	0	Pad up to 2-byte boundary

EPI-Lo executes commands, either from its host or from command sequences stored locally. Most commands are executed as soon as they are received; exceptions are noted in the individual command descriptions. Each command is echoed after it is executed. A command echo packet containing the opcode and some arguments along with a code summarizing the command's result is downlinked. If the command's opcode is unknown, is not accompanied by a sufficient number of arguments, or contains invalid arguments, the command will be rejected. The command echo packet gives the reason for rejection and the commands-rejected counter is incremented. If the command is executed successfully, the command echo indicates this and the commands-executed counter is incremented. Commands being executed from a macro undergo similar verification. There are a corresponding pair of executed and rejected counters for commands from macros.

7.2.1 Macro Processing

A macro is a sequence of commands that can be stored, then executed later. 64 KiB of processor memory are reserved for the storage of macros. A macro can be any length as long as the total length of all the macros fit in the memory. Each command stored in a macro is constrained to have at most 253 arguments. Up to 256 macros may be defined; each macro is identified by a small integer, 0 – 255.

The command handler can be commanded into a "learning" mode to define a macro. Each command has a "macro" field indicating whether the command should be executed or added to a macro. If a command arrives with its macro field indicating "learn" when the command handler is in learning mode, the command is echoed, but not executed; instead the command is added to the macro currently being constructed. Any command that arrives without the "learn" macro field, will be handled as usual. When not in learn mode, a command that arrives with a "learn" macro field will be rejected. Later, when a macro is executed, its component commands will be executed and echoed.

There are commands for running and stopping a macro. There are also special commands for adding delays to a macro; these can be used only within a macro. One macro can call another or can execute loops. Both calls and loops can be nested. Loop nesting depth is traded with macro nesting depth. Each running macro context has a 32 element stack. A macro call uses two elements and a loop uses three elements.

Up to 64 macros can be running concurrently. If more than one macro has been commanded to run, one of them is run until completion or until a delay command is executed. Then the next runnable macro is run. At all times, new commands arriving from the host system have priority over macros. Since all commands are executed as soon as they are received from the host system, a macro run command and the macro's constituent commands will also be run immediately. However, the first macro command after a macro delay or pause command will execute soon after the occurrence of a 1 Hz time pulse.

7.2.2 Default Macros

Several default macros are available as soon as EPI-Lo is started. Most of these macros are one or two commands long and are used by the monitoring subsystem to respond to out of limit conditions. A default macro can be replaced with a new macro via EPILO_MAC_DEF and EPILO_MAC_ENDDEF.

Table 21. Default Macros

ID	Action	Commands
0	No action	EPILO_CMD_NULL
1	Shutdown	TBD
2	Safing	TBD
3	Startup	TBD
4	MCP HV A over-current	TBD
5	MCP HV B over-current	TBD
6	MCP HV C over-current	TBD
7	MCP HV D over-current	TBD
8 - 32	*Reserved for operations autonomy*	
33 – 255	*Reserved*	

7.3 Core Commands

The EPI-Lo core commands are defined below.

7.3.1 EPILO_CMD_NULL – Do Nothing

Do nothing. Useful for testing command uplink and command echo downlink.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x01	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.2 EPILO_CMD_TEXT – Send Text

Downlink the text string argument.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3 - 79	Packet length (bytes) - 1
Opcode	8	0x21	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Byte Count	8	0 - 77 (C)	Number of characters
Text	C * 8		Character data
Pad	0 or 8	0	Pad up to 2-byte boundary

7.3.3 EPILO_CMD_WRAP – Wrapped Command

Process the arguments as a command. Note: the wrapped command will be echoed, not the EPILO_CMD_WRAP command.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3 - 265	Packet length (bytes) - 1
Opcode	8	0x02	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Opcode	8		Command opcode
Arguments	M * 8		Arguments
Pad	0 or 8	0	Pad up to 2-byte boundary

7.3.4 EPILO_MAC_CHECK – Check Macros

Calculate the checksums of a range of macros. The checksum is the 16-bit CRC-CCITT of the macro. The checksums are downlinked in a packet.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x03	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
First Macro	8	0 – 255	Id of first macro to check
Last Macro	8	0 – 255	Id of last macro to check

7.3.5 EPILO_MAC_DEF – Begin Macro Definition

Start defining the identified macro. All subsequent commands, which have a "learn" macro field, will be added to the macro; commands without a "learn" macro field, will be executed as usual. Macro definition continues until a EPILO_MAC_ENDDEF command is seen; EPILO_MAC_ENDDEF must have an "execute" macro field to end the definition. The macro cannot be run until the definition is complete. If a macro with the same id already exists, the new macro will replace the old when the EPILO_MAC_ENDDEF is received. The command will be rejected if the id is invalid or a macro is already being defined.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x04	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.3.6 EPILO_MAC_DELAY – Delay Macro

Delay macro execution. When a macro executes this command, it will stop for the given number of seconds. This command will be rejected if used outside of a macro definition.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x05	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control
Delay	16	Unsigned integer	Delay (seconds)

7.3.7 EPILO_MAC_END – End Macro

Terminate macro execution. If this is a nested macro, resume the calling macro. This command will be rejected if used outside of a macro definition. Note: the user never needs to use this command explicitly; see EPILO_MAC_ENDDEF below.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x06	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control

7.3.8 EPILO_MAC_ENDDEF – End Macro Definition

Terminate macro definition. Add an EPILO_MAC_END command to the macro being defined and end the definition. This command will be rejected if no macro is being defined.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x07	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.9 EPILO_MAC_HALT – Halt Macro

Stop the identified macro. This command will be rejected if the id is invalid, if the identified macro is undefined, or it is not running.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x08	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.3.10 EPILO_MAC_LOOP_BEG – Start a Macro Loop

Start a loop with the given number of iterations. EPILO_MAC_LOOP_END decrements the loop index. If it is non-zero, the next command to be executed will be the one after the EPILO_MAC_LOOP_BEG. If it is zero, the next command will be the one after the EPILO_MAC_LOOP_END. This command will be rejected if used outside of a macro definition.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x09	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control
Iterations	16	Unsigned integer	Loop iterations

7.3.11 EPILO_MAC_LOOP_END – End a Macro Loop

Mark the end of a macro loop (see EPILO_MAC_LOOP_BEG). This command will be rejected if used outside of a macro definition.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x0a	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control

7.3.12 EPILO_MAC_NEST – Nest Macro

Run the identified macro. The macro that contains the EPILO_MAC_NEST command is suspended until the new macro completes. This command will be rejected if the id is invalid or if the identified macro is undefined. This command will also be rejected if used outside of a macro definition.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x0b	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.3.13 EPILO_MAC_PAUSE – Pause Macro

Pause macro execution. When a macro executes this command, the macro will pause until the actual time is greater than or equal to the given time. This command will be rejected if used outside of a macro definition.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x0c	Opcode
Macro?	8	0 = Execute (invalid) 1 = Append to macro	Macro learn control
Time	32	Unsigned integer	Time (seconds)

7.3.14 EPILO_MAC_READ – Read Macros

Dump the definitions of a range of macros. One dump packet is generated each second.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x0d	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
First Macro	8	0 – 255	Id of first macro to read
Last Macro	8	0 – 255	Id of last macro to read

7.3.15 EPILO_MAC_RESTORE – Restore Macros Saved in Non-Volatile Memory

Restore the macros saved in non-volatile memory. All running macros are aborted. The command will be rejected if the checksum of the saved macros is invalid.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x0e	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.16 EPILO_MAC_RUN – Run Macro

Run the identified macro. Command echoes will be generated for commands in the macro and in all nested macros. This command will be rejected if the id is invalid or if the identified macro is undefined.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x0f	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.3.17 EPILO_MAC_RUN_SIL – Run Macro Silently

Run the identified macro, but suppress command echoes for commands in the macro and in all nested macros. This command will be rejected if the id is invalid or if the identified macro is undefined.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x10	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.3.18 EPILO_MAC_SAVE – Save Macros to Non-Volatile Memory

Save all macros to non-volatile memory.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x11	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.19 EPILO_MEM_CHECK – Check Memory

Calculate the checksum of up to 65536 bytes of memory at the given address. The checksum is the 16-bit CRC-CCITT of the memory indicated. The results are downlinked in a packet. Note that the amount of memory to check is specified in 16-bit words.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	7	Packet length (bytes) - 1
Opcode	8	0x12	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Source	32	Unsigned integer	Source address
Count	16	Unsigned integer	Number of 16-bit words to check

7.3.20 EPILO_MEM_COPY – Copy Memory

Copy processor memory from source to destination. The source or destination can be either RAM or MRAM. The result is undefined if the source and destination regions overlap. This command will be rejected unless memory write operations are enabled via the EPILO_MEM_WR_EN command.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	11	Packet length (bytes) - 1
Opcode	8	0x13	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Source	32	Unsigned integer	Source address
Destination	32	Unsigned integer	Destination address
Byte Count	16	Unsigned integer	Number of bytes to copy

7.3.21 EPILO_MEM_LOAD – Load Memory

Load processor RAM or MRAM memory. The byte count determines how many load bytes follow and may be up to 256 bytes. This command will be rejected unless memory write operations are enabled via the EPILO_MEM_WR_EN command.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9 - 265	Packet length (bytes) - 1
Opcode	8	0x14	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Address	32	Unsigned integer	Starting address
Byte Count	16	0 – 256 (M)	Number of bytes in data field
Reserved	16		
Data	M * 8		Load data
Pad	0 or 8	0	Pad up to 2-byte boundary

7.3.22 EPILO_MEM_READ – Read Memory

Dump up to 65536 bytes of memory from given address. The dump length is rounded up to a multiple of four bytes. One dump packet is generated each second. If a dump is in progress and a new EPILO_MEM_READ command is issued, the original dump is aborted and a new dump begins.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9	Packet length (bytes) - 1
Opcode	8	0x15	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Source	32	Unsigned integer	Source address
Unused	16		
Byte Count	16	Unsigned integer	Number of bytes to dump

7.3.23 EPILO_MEM_READ_ABT – Abort Memory Read

Abort the current memory dump.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x16	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.24 EPILO_MEM_RUN – Run a Program

Run a program at the given address. The program must either execute very quickly (so as not to hold up the command handler) or must take over completely. If the program returns, there must be no net stack effects.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x17	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Address	32	Unsigned integer	Program start address

7.3.25 EPILO_MEM_STR_LOAD – Load Values into a Data Structure

Load the data values into the identified data structure at the location denoted by the offset. The byte count determines how many load bytes follow and may be up to 255 bytes.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5 - 261	Packet length (bytes) - 1
Opcode	8	0x18	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Id	8	0 = Monitor limits 1 = Parameters	Id of data structure
Byte Count	8	0 – 255 (M)	Number of bytes in data field
Offset	16	Unsigned integer	Byte offset into data structure
Data	M * 8		Load data
Pad	0 or 8	0	Pad up to 2-byte boundary

7.3.26 EPILO_MEM_STR_READ – Read a Data Structure

Dump the identified data structure.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x19	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Id	8	0 = Monitor limits 1 = Parameters	Id of data structure
Pad	8	0	Pad up to 2-byte boundary

7.3.27 EPILO_MEM_WR_EN – Enable / Disable Memory Write

Enable or disable memory writes. If disabled, EPILO_MEM_LOAD and EPILO_MEM_COPY commands will be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x1a	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable memory write.
Pad	8	0	Pad up to 2-byte boundary

7.3.28 EPILO_MON_CNTRL – Enable / Disable Monitor Response

Enable or disable monitor response. If enabled, a persistent out-of-limits condition will cause a corresponding response macro to be executed.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x1b	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable response
Pad	8	0	Pad up to 2-byte boundary

7.3.29 EPILO_SAF_CYCLE – Request Power Cycle

Request that the spacecraft turn off EPI-Lo's power, then turn it back on.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x25	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.30 EPILO_SAF_OFF – Request Power Off

Request that the spacecraft turn off EPI-Lo's power.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x22	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.31 EPILO_SAF_RESET – Initiate Software Reset

Reset the EPI-Lo instrument by allowing the watchdog to go off. Note: the command will not be echoed.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x23	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

7.3.32 EPILO_SAF_TIMEOUT – Set Safing Timeout

Set spacecraft to EPI-Lo communication timeout interval.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x24	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Time	16	0 – 65535	Timeout interval (seconds)

7.3.33 EPILO_STAT_CLR – Clear Counter

Clear selected counter in status packet.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x1c	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Counter	8	0 = Commands executed 1 = Commands rejected 2 = Macro commands executed 3 = Macro commands rejected 4 = Telemetry volume 5 = Alarm count 255 = All	Counter to clear
Pad	8	0	Pad up to 2-byte boundary

7.3.34 EPILO_STAT_INT – Set Status Generation Interval

Set the interval between status packets.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x1d	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Interval	16	1 – 65535 (0 = Off)	Interval (seconds)

7.3.35 EPILO_STAT_MEM – Select Status Memory Monitor

Select the address of the memory region to be included in the status packet.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x20	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Address	32	Unsigned integer	Source address

7.4 EPI-Lo Application Commands

The EPI-Lo application commands are defined below. Most of the commands take effect immediately. Exceptions are noted below.

7.4.1 EPILO_BV_ENB – Enable / Disable SSD Bias Voltage

Enable or disable SSD bias voltage power supply.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x57	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable SSD bias voltage
Pad	8	0	Pad up to 2-byte boundary

7.4.2 EPILO_BV_LEVEL – Set SSD Bias Voltage Goal

Set the goal of the SSD bias voltage power supply. The bias voltage power supply must be enabled (EPILO_BV_ENB) before the level can be set; otherwise, the command will be rejected. A command to set the goal above its limit (EPILO_BV_LIMIT) will also be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x58	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Goal	8	0 - 255	SSD bias voltage goal
Pad	8	0	Pad up to 2-byte boundary

7.4.3 EPILO_BV_LIMIT – Set SSD Bias Voltage Limit

Set the upper limit of the SSD bias voltage power supply. A command to set the limit below its goal (EPILO_BV_LEVEL) will be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x59	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Limit	8	0 - 255	SSD bias voltage limit
Pad	8	0	Pad up to 2-byte boundary

7.4.4 EPILO_BV_STEP – Step SSD Bias Voltage Goal

Step the goal of the SSD bias voltage power supply. The bias voltage power supply must be enabled (EPILO_BV_ENB) before the level can be set; otherwise, the command will be rejected. A command to set the goal above its limit (EPILO_BV_LIMIT) will also be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5a	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Step	16	-255 - 255	SSD bias voltage goal step

7.4.5 EPILO_DAT_COLLECT – Set Data Collection Pattern

Set the pattern of hardware modes used in each slot and thus what data is collected and in what proportion. The command takes effect on the next slow (T*N1*N2) integration.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9	Packet length (bytes) - 1
Opcode	8	0x52	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Slot 0	8	0 = Ion composition 1 = Ion energy 2 = Particle comp. 3 = Particle energy	Data collection pattern
Slot 1	8		
Slot 2	8		
Slot 3	8		
Slot 4	8		
Slot 5	8		
Slot 6	8		
Slot 7	8		

7.4.6 EPILO_DAT_ENB – Enable / Disable Data Products

Enable or disable collection and telemetry of selected data product. The command takes effect at the start of the next slow (T*N1*N2) integration. TBD: burst data here, or in its own command?

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x53	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable product
Product	8	0 = Ion comp. basic 1 = Ion comp. diag. 2 = High-res. protons 3 = High time-res. protons 4 = Ions1 5 = Ions2 6 = PH scan 7 = Mass histograms 8 = Ion TOF 9 = Ion energy basic 10 = Ion energy diag. 11 = Ion energy 12 = Particle comp. basic 13 = Particle comp. diag. 14 = Particles 15 = Particle TOF 16 = Particle energy basic 17 = Particle energy diag. 18 = High-res. electrons 19 = High time-res. electrons 20 = High look-res electrons	Data product

Name	Length (bits)	Value	Description
		21 = High-energy particles 22 = All	

7.4.7 EPILO_DAT_EVT – Control Amount of Event Data

Set the maximum number of events to telemeter per integration. Note: setting the number of events to zero disables the data product. The command takes effect at the start of the next slow (T*N1*N2) integration.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x54	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Events	8	0 - TBD	Maximum number of events
Product	8	0 = Ion comp. raw 1 = Ion energy raw 2 = Part. comp. raw 3 = Part. energy raw 4 = Ion comp. priority 5 = Part. comp. priority 6 = Part. energy priority 7 = All	Data product

7.4.8 EPILO_DAT_TIME – Control Data Integration Time

Control time over which to integrate data products. Fast data products are integrated for T seconds. Medium data products are integrated for T*N1 and T*N2 seconds. Slow data products are integrated for T*N1*N2 seconds. The command will be rejected if the slow integration time exceeds 65535 seconds. The command takes effect at the end of the current fast (T) integration; medium and slow integrations are aborted and their products telemetered.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x55	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Integration Time	8	1 - 255	Integration time, T (seconds)
Integration Mult. N1	8	1 - 255	Integration multiplier, N1
Integration Mult. N2	8	1 - 255	Integration multiplier, N2
Pad	8	0	Pad up to 2-byte boundary

7.4.9 EPILO_EGY_ENERGY – Set Energy Threshold

Set selected ion or particle energy threshold.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x40	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Threshold	8	0 - 255	Energy threshold
SSD	8	0 = Ion 1 = Particle	
Channel	8	0 = A1 1 = A2 2 = B1 3 = B2 4 = C1 5 = C2 6 = D1 7 = D2	Channel
Pad	8	0	Pad up to 2-byte boundary

7.4.10 EPILO_EGY_EXTEND – Set Extended Energy Threshold

Set selected ion or particle extended energy threshold.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x41	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Threshold	8	0 - 255	Extended energy threshold
SSD	8	0 = Ion 1 = Particle	
Channel	8	0 = A1 1 = A2 2 = B1 3 = B2 4 = C1 5 = C2 6 = D1 7 = D2	Channel
Pad	8	0	Pad up to 2-byte boundary

7.4.11 EPILO_EVT_MULTI – Enable / Disable Multiple Hit Reject

Enable or disable multiple hit rejection on energy detector.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x46	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable multiple hit rejection
Pad	8	0	Pad up to 2-byte boundary

7.4.12 EPILO_EVT_TDC_CHECK – Enable / Disable TDC Consistency Check

Enable or disable TDC consistency check. If enabled, the TDC values are required to be consistent with each other.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x47	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable TDCs consistency check
Pad	8	0	Pad up to 2-byte boundary

7.4.13 EPILO_EVT_TDC_MATCH – Enable / Disable TDC and SSD Match

Enable or disable TDC/SSD position match check. If enabled, the position calculated by the TDC is required to match the SSD channel.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x48	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable TDC and SSD match
Pad	8	0	Pad up to 2-byte boundary

7.4.14 EPILO_EVT_TDC_ONLY – Enable / Disable TDC-Only Events

Enable or disable TDC-only events, i.e. events with no SSD energy measurement. If enabled, TDC-only ion composition or particle composition events may be placed in the event FIFO. Enable options allow all events to be saved or some events to be saved, e.g. 1 of 4 events.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x4a	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable all 2 = Enable 1 of 2 3 = Enable 1 of 4 4 = Enable 1 of 8 5 = Enable 1 of 16 6 = Enable 1 of 32 7 = Enable 1 of 64	Enable/disable TDC-only events
Pad	8	0	Pad up to 2-byte boundary

7.4.15 EPILO_EVT_TDC_POS – Enable / Disable TDC Position Valid

Enable or disable TDC position validity check. If enabled, the position calculated by the TDC is required to be valid.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x49	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable TDC position valid
Pad	8	0	Pad up to 2-byte boundary

7.4.16 EPILO_HV_COM_LEVEL – Set Common High Voltage Goal

Set the goal of the common high voltage power supply. The high voltage power supply must be enabled (EPILO_HV_ENB) before the level can be set; otherwise, the command will be rejected. A command to set the goal above its limit (EPILO_HV_COM_LIMIT) will also be rejected. A command that violates the constraints with individual MCP HV goals (see 4.1) will be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5b	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Goal	8	0 - 255	High voltage goal
Pad	8	0	Pad up to 2-byte boundary

7.4.17 EPILO_HV_COM_LIMIT – Set Common High Voltage Limit

Set the upper limit of the common high voltage power supply. A command to set the limit below its goal (EPILO_HV_COM_LEVEL) will be rejected. If the safing plug is installed, the command will be rejected if the limit is above the safe limit.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5c	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Limit	8	0 - 255	High voltage limit
Pad	8	0	Pad up to 2-byte boundary

7.4.18 EPILO_HV_CUR_ENB – Enable / Disable MCP HV Current Monitor

Enable or disable hardware current monitoring and automatic shutdown of the selected MCP HV supply.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x50	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable current monitor
Quadrant	8	0 = A 1 = B 2 = C 3 = D	Quadrant

7.4.19 EPILO_HV_CUR_LIMIT – Set MCP HV Current Monitor Limit

Set the upper current limit of the selected MCP HV supply.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x51	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Limit	8	0 - 255	Current monitor limit
Quadrant	8	0 = A 1 = B 2 = C 3 = D	Quadrant

7.4.20 EPILO_HV_ENB – Enable / Disable High Voltage

Enable or disable high voltage power supply.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5d	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Enable/disable high voltage
Pad	8	0	Pad up to 2-byte boundary

7.4.21 EPILO_HV_MCP_LEVEL – Set MCP High Voltage Goal

Set the goal of the selected MCP high voltage power supply. The high voltage power supply must be enabled (EPILO_HV_ENB) before the level can be set; otherwise, the command will be rejected. A command to set the goal above its limit (EPILO_HV_MCP_LIMIT) will also be rejected. A command that violates the constraints with the common HV goal (see 4.1) will be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5e	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Goal	8	0 - 255	MCP high voltage goal
Quadrant	8	0 = A 1 = B 2 = C 3 = D	Quadrant

7.4.22 EPILO_HV_MCP_LIMIT – Set MCP High Voltage Limit

Set the upper limit of the selected MCP high voltage power supply. A command to set the limit below its goal (EPILO_HV_MCP_LEVEL) will be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x5f	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Limit	8	0 - 255	MCP high voltage limit
Quadrant	8	0 = A 1 = B 2 = C 3 = D	Quadrant

7.4.23 EPILO_HV_MCP_STEP – Step MCP High Voltage Goal

Step the goal of the selected MCP high voltage power supply. The high voltage power supply must be enabled (EPILO_HV_ENB) before the level can be set; otherwise, the command will be rejected. A command to set the goal above its limit (EPILO_HV_MCP_LIMIT) will also be rejected.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x60	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Voltage Step	16	-255 - 255	MCP high voltage goal step
Quadrant	8	0 = A 1 = B 2 = C 3 = D	Quadrant
Pad	8	0	Pad up to 2-byte boundary

7.4.24 EPILO_OP_INIT – Initialize Autonomous Operations

Initialize autonomous operations. The autonomous operations region definitions are all deleted; new regions can be defined by command (EPILO_OP_REGION). If no regions are defined and autonomous operations are started, the commanded macro is run.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x44	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Pad	8	0	Pad up to 2-byte boundary

7.4.25 EPILO_OP_REGION – Define Autonomous Operations Region

Define an autonomous operations region. Once the spacecraft enters the region of the orbit defined by the commanded solar distance, the commanded macro is run. The command is rejected if the maximum number of regions have been defined.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	7	Packet length (bytes) - 1
Opcode	8	0x45	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Macro Id	8	0 – 255	Macro id
Solar Distance Inflection	8	0 = Decreasing 1 = Increasing	Solar distance, i.e. distance from spacecraft to the sun, and whether distance is decreasing or increasing
Solar Distance	32	Unsigned integer	

7.4.26 EPILO_TDC_CFD – Set TDC System CFD Threshold

Set selected TDC system CFD threshold.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x42	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Threshold	16	0 - 511	Anti-coincidence threshold
Channel	8	0 = A Start1 1 = A Start2 2 = A Stop 3 = B Start1 4 = B Start2 5 = B Stop 6 = C Start1 7 = C Start2 8 = C Stop 9 = D Start1 10 = D Start2 11 = D Stop	Channel
Pad	8	0	Pad up to 2-byte boundary

7.4.27 EPILO_TDC_PH – Set TDC System Pulse Height Threshold

Set selected TDC system pulse height threshold.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x43	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Threshold	8	0 - 255	PH threshold
Channel	8	0 = A and B 1 = C and D	Channel

7.4.28 EPILO_TDC_PH_STEP – Step TDC System Pulse Height Thresholds

Step both TDC system pulse height thresholds. If a new threshold exceeds its allowed range, the threshold is clamped to the closest value within range.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x56	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Step	16	-255 - 255	PH threshold step

7.4.29 EPILO_TST_BASE – Measure Energy Baseline

Measure baseline of selected SSD energy channel. Note: baseline measurement suspends normal event processing in the FPGA.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x4b	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Channel	8	0 = A1 1 = A2 2 = B1 3 = B2 4 = C1 5 = C2 6 = D1 7 = D2 8 = Disable	Channel to measure, or disable
Pad	8	0	Pad up to 2-byte boundary

7.4.30 EPILO_TST_PUL_CFG – Configure Internal Pulser

Configure internal pulser. Sets event rate and delays. Note: selecting the external source is only used for testing on the ground.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x4c	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Rate	8	0 = ~300 Hz 1 = 400 Hz 2 = 500 Hz 3 = ~600 Hz 4 = ~700 Hz 5 = 800 Hz 6 = ~900 Hz 7 = 1 kHz 8 = 2 kHz 9 = ~3 kHz 10 = 4 kHz 11 = 5 kHz 12 = ~6 kHz 13 = ~7 kHz 14 = 10 kHz 15 = 25 kHz	Event rate
Start to Stop Delay	8	0 - 15	Delays (12.5 ns units)
Start to SSD1 Delay	8	0 - 63	
Start to SSD2 Delay	8	0 - 63	

7.4.31 EPILO_TST_PUL_ENB – Enable / Disable Internal Pulser

Enable or disable selected pulse destination. Note: if all destinations are disabled, the pulser circuit is turned off.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x4d	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Mode	8	0 = Disable 1 = Enable	Pulse enable/disable
Destination	8	0 = A Start 1 = A Stop 2 = B Start 3 = B Stop 4 = C Start 5 = C Stop 6 = D Start 7 = D Stop 8 = SSD1 9 = SSD2	Pulse destination

7.4.32 EPILO_TST_PUL_HGT – Set Internal Pulse Height

Set height of internal pulses destined for the anodes or the SSDs.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x4e	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Height	8	0 - 255	Pulse height
Channel	8	0 = Anode 1 = SSD	Channel

7.4.33 EPILO_TST_SIGNAL – Select Test Point Signal

Select signal to bring out on indicated test point. Note: this command is only used for testing on the ground.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3	Packet length (bytes) - 1
Opcode	8	0x4f	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Signal	8	0 - 3 (0 = 0V)	Signal to test
Test Point	8	0 - 4	Test point

8. Telemetry Interface

EPI-Lo sends telemetry ITFs to the spacecraft. A frame can be at most 8196 bytes, including all headers; the EPI-Lo software sends frames that are at most 2060 bytes. The length of the telemetry ITF must be an even number of bytes. During each second EPI-Lo accumulates packets into a transfer frame. Near the end of the second, EPI-Lo freezes the transfer frame and prepares to send it during the next second. Telemetry frame transfer starts at the 1PPS and will complete before the next 1PPS. The frame is variable length and can contain one or more packets; the length field tells how much packet data is present. The first header field identifies the position of the first packet in the frame; this allows a packet to straddle frames. If a packet straddles more than two frames, at least one frame will have no packet headers; this is encoded with a special value (0xffff) in the first header field. If EPI-Lo has no packets to deliver, the length field will have the minimum value and the first header field will be 0xffff. The telemetry ITF format is shown in the following table.

Table 22. Telemetry ITF Format

Name	Length (bits)	Value	Description
Sync	4 * 8	0xfefa30c8	Synchronization pattern
Virtual Channel	8	0x05	Fixed value for each instrument interface
Sequence	8	Unsigned integer	Sequence number per virtual channel
Length	16	4 - 2052	Number of bytes following the Length field, including the Checksum
First Header	16	Unsigned integer (0xffff = no header)	Offset of first packet header in the Data
Data	N * 8		Telemetry data
Checksum	16		16-bit exclusive-or of ITF, except for sync

8.1 Packet Telemetry

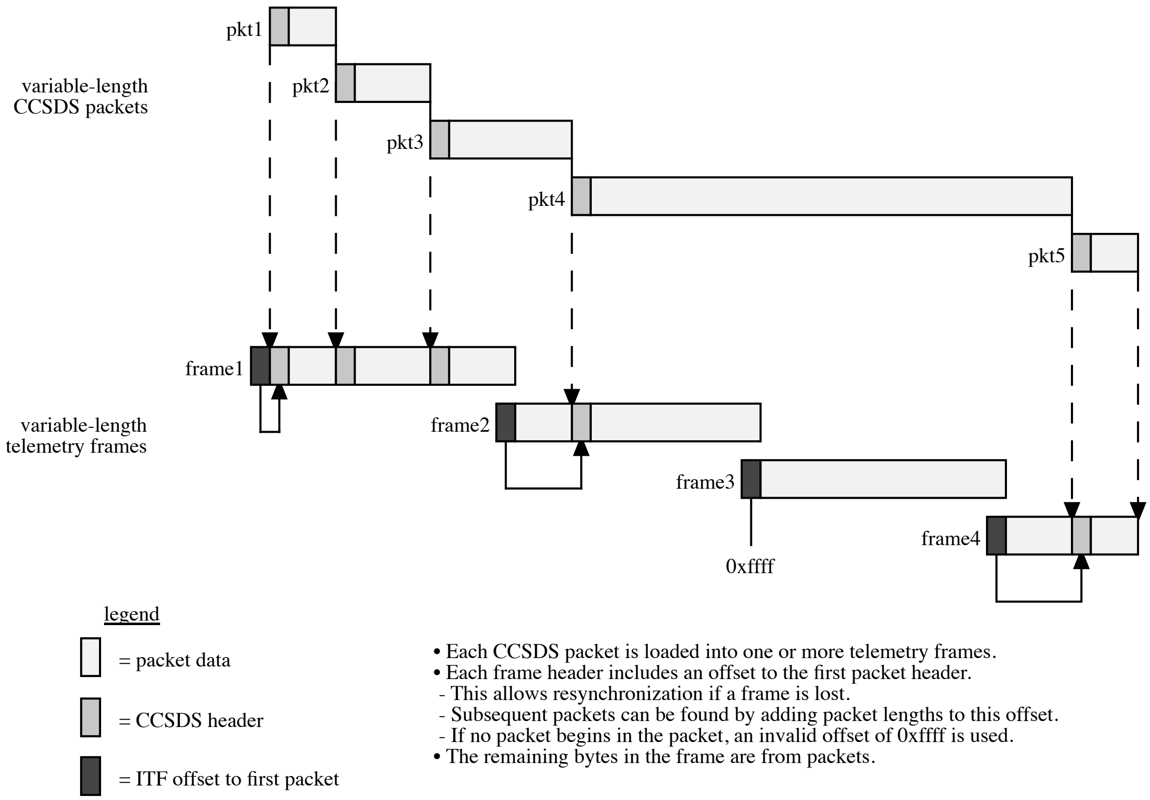
EPI-Lo produces variable-sized CCSDS telemetry packets. Each packet has a header containing an application process identifier (APID), length, grouping flags, sequence count, and time-tag. The packet APID identifies the type of data. The grouping flags allow a series of packets to be treated as a unit in ground processing. The time tag in the packet corresponds to when the packet data was sampled. The length is the number of bytes in the data portion of the packet. The pad rounds the entire length up to a multiple of 2 bytes. The packet may be up to 4096 bytes, including headers.

Table 23. Telemetry Packet Format

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	0x4a0 - 0x4df	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	3 - 4089	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Data	N * 8		Data payload
Pad	0 - 8	0	Pad up to 2-byte boundary

The following figure illustrates how EPI-Lo packets are mapped to telemetry frames. Packets may be split across multiple frames. In the figure, packet 3 is split across two frames and packet 4 is split across three frames.

Figure 25. EPI-Lo Packets



The following table lists the various packets, their APIDs, and how often they are sent.

Table 24. Packet Types

Product	APID	When Sent
Reserved	0x4a0	
Command Echo	0x4a1	As needed
Alarm	0x4a2	As needed
Memory Checksum	0x4a3	On demand
Memory Dump	0x4a4	On demand, one packet per second
Status	0x4a5	Commandable interval
Boot Status	0x4a6	Commandable interval
Macro Dump	0x4a7	On demand, one packet per second
Macro Checksums	0x4a8	On demand
Monitor Limits	0x4a9	On demand
Parameters	0x4aa	On demand
Text	0x4ab	On demand
Critical Housekeeping	0x4ac	Every second
Reserved	0x4ad – 0x4af	
Ion Composition Basic Rates	0x4b0	Every commanded number of seconds (see Table 4), if enabled
Ion Composition Diagnostic Rates	0x4b1	
High-Res. Protons	0x4b2	
High Time-Res. Protons	0x4b3	
Ions1	0x4b4	
Ions2	0x4b5	
PH Scan	0x4b6	
Mass Histograms	0x4b7	
Ion TOF	0x4b8	
Ion Energy Basic Rates	0x4b9	
Ion Energy Diagnostic Rates	0x4ba	
Ion Energy	0x4bb	
Particle Composition Basic Rates	0x4bc	
Particle Composition Diag. Rates	0x4bd	
Particles	0x4be	
Particle TOF	0x4bf	
Particle Energy Basic Rates	0x4c0	
Particle Energy Diagnostic Rates	0x4c1	
High-Res. Electrons	0x4c2	
High Time-Res. Electrons	0x4c3	
High Look-Res Electrons	0x4c4	
High-Energy Particles	0x4c5	
Raw Ion Composition	0x4c6	

Raw Ion Energy	0x4c7	
Raw Particle Composition	0x4c8	
Raw Particle Energy	0x4c9	
Priority Ion Composition	0x4ca	
Priority Particle Composition	0x4cb	
Priority Particle Energy	0x4cc	
Reserved	0x4cd - 0x4df	

The formats of the different packets are described below.

8.2 Data Compression

EPI-Lo compresses large values using a lossy logarithmic algorithm. Values may be further losslessly compressed using a "Fast" algorithm.

8.2.1 32-bit to 10-bit Log Compression Algorithm

The EPI-Lo flight software employs a 32-bit to 10-bit log compression for software counters, histograms, etc. The following table shows the log compressed format. Both e and m are 5-bit unsigned integers; e is always less than 28.

Table 25. 10-bit Log Compressed Format

e	m
5-bits	5-bits

Decompression is done using the following pseudo-C algorithm:

```
if e==0
      return m
else
      return (32+m) << (e-1)
```

8.2.2 Fast Compression

Some arrays of SIS data are compressed using a lossless "Fast" compression algorithm. Each array value is 10 bits (after log compression). The values are organized in blocks of 16 values. Each block is compressed as a unit; if the length of the array is not a multiple of 16, the final unit will have fewer values. The maximum value determines how many bits are used for each value. Each block is sent with a width field (the number of bits needed to per values) and 16 (or possibly fewer in the last block) bit-packed values.

Table 26. Fast Compression Format

Name	Length (bits)	Value	Description
Width 1	4	0 - 10	Block 1 with width (W1) and 16 bit-packed values
Values 1	16 * W1	Unsigned Integers	
Width 2	4	0 - 10	Block 2 with width (W2) and 16 bit-packed values
Values 2	16 * W2	Unsigned Integers	
…			
Width N	4	0 - 10	Block N with width (WN) and up to 16 bit-packed values
Values N	1-16 * WN	Unsigned Integers	

Decompression is done using the following pseudo-code algorithm:

```
for each block
      width = read(4 bits)
      for each value in block /* 16 values, 1-16 in last */
            Vi = read(width bits)
Where read(n) is a function that reads next n bits; read(0) returns 0.
```

8.3 Engineering Packets

The following sections describe the EPI-Lo engineering packets.

8.3.1 Command Echo

Each command received is echoed in a packet. Each echo packet consists of a header and command echo data. The grouping flags in the header will be set to “none” and the time tag will be set to the time at which the command was executed. A macro bit (set to 1 if the command was executed as part of a macro) and a code summarize the command’s result. The result codes are defined in Appendix 2. The command’s opcode and up to ten argument bytes appear in the packet’s data; this is a variable-length packet.

Table 27. Command Echo Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5 - 15	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Macro?	1	0 = Uplink 1 = Macro	Set if executed in a macro
Result	7	See Table 73	Command result code
Opcode	8	Unsigned integer	Command opcode
Arguments	N * 8		Up to ten argument bytes
Pad	0 or 8	0	Pad up to 2-byte boundary

8.3.2 Alarm

Alarm packets report problems. Each alarm packet consists of a header and alarm data. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the problem occurred. The alarm data consists of a byte identifying the alarm's cause and two value bytes with additional information. The type indicates transient or persistent alarms.

Table 28. Alarm Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	7	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Alarm Id	8	See Table 72	Alarm identifier
Type	8	0 = Persistent 1 = Transient	Alarm type
Value	8		Value associated with alarm
Auxiliary	8		Another value associated with alarm

8.3.3 Memory Checksum

Memory checksum packets report a checksum over a given memory region. Each checksum packet consists of a header and checksum data. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the checksum was computed. The checksum is the 16-bit CRC-CCITT of the indicated memory region.

Table 29. Memory Checksum Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	11	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Address	32	Unsigned integer	Address of region checked
Length	16	Unsigned integer	Length of region checked (words)
Checksum	16	Unsigned integer	Computed checksum

8.3.4 Memory Dump

Memory dump packets telemeter a selected range of memory contents. Each dump packet consists of a header and memory data. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the dump packet was generated. The dump data consists of a dump start address, dump length, and up to 256 bytes of dump data; this is a variable length packet. Note: the dump length is always rounded up to a multiple of four bytes to avoid having to pad the end of the packet with zeros.

Table 30. Memory Dump Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	15 - 267	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Address	32	Unsigned integer	Starting address
Length	16	4 – 256 (N)	Data length (bytes)
Reserved	16	0	
Data	N * 8		Dump data

8.3.5 Status

EPI-Lo generates a status packet periodically. The rate is controlled by a command (see EPILO_STAT_INT). Each status packet consists of a header and status data; the status includes analog, digital, and software data. The grouping flags in the header will be set to “none” and the time tag will be set to the time at which the status packet was generated. The overall packet format is shown in the following table; the formats of the individual components are shown in the subsequent tables.

Table 31. Status Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	TBD	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Analog	60 * 8	See Table 32	Analog readings
Digital	82 * 8	See Table 33	Digital configuration and feedback
Software	54 * 8	See Table 34	Software configuration and status

Table 32. Analog Status

Name	Length (bits)	Value	Description
ADC/0/0	12	0 - 4095	A1 SSD temperature
ADC/0/1	12	0 - 4095	A2 SSD temperature
ADC/0/2	12	0 - 4095	B1 SSD temperature
ADC/0/3	12	0 - 4095	B2 SSD temperature
ADC/0/4	12	0 - 4095	C1 SSD temperature
ADC/0/5	12	0 - 4095	C2 SSD temperature
ADC/0/6	12	0 - 4095	D1 SSD temperature
ADC/0/7	12	0 - 4095	D2 SSD temperature
ADC/1/0	12	0 - 4095	A Anode temperature
ADC/1/1	12	0 - 4095	C Anode temperature
ADC/1/2	12	0 - 4095	C Anode temperature
ADC/1/3	12	0 - 4095	D Anode temperature
ADC/1/4	12	0 - 4095	Event board temperature 1
ADC/1/5	12	0 - 4095	Event board temperature 2
ADC/1/6	12	0 - 4095	*Spare*
ADC/1/7	12	0 - 4095	*Spare*
ADC/2/0	12	0 - 4095	+1.5V current
ADC/2/1	12	0 - 4095	+3.3V current
ADC/2/2	12	0 - 4095	+5V current
ADC/2/3	12	0 - 4095	+13V current
ADC/2/4	12	0 - 4095	LVPS temperature
ADC/2/5	12	0 - 4095	Primary current
ADC/2/6	12	0 - 4095	*Ground/Spare*
ADC/2/7	12	0 - 4095	*Ground/Spare*
ADC/3/0	12	0 - 4095	+3.5V voltage
ADC/3/1	12	0 - 4095	+13V voltage
ADC/3/2	12	0 - 4095	+5V voltage
ADC/3/3	12	0 - 4095	+3.3V voltage
ADC/3/4	12	0 - 4095	+1.5V voltage
ADC/3/5	12	0 - 4095	SSD BV power
ADC/3/6	12	0 - 4095	SSD BV voltage
ADC/3/7	12	0 - 4095	Common HV voltage
ADC/4/0	12	0 - 4095	A HV voltage
ADC/4/1	12	0 - 4095	B HV voltage
ADC/4/2	12	0 - 4095	C HV voltage
ADC/4/3	12	0 - 4095	D HV voltage
ADC/4/4	12	0 - 4095	A HV current
ADC/4/5	12	0 - 4095	B HV current

Name	Length (bits)	Value	Description
ADC/4/6	12	0 - 4095	C HV current
ADC/4/7	12	0 - 4095	D HV current

Table 33. Digital Status

Name	Length (bits)	Value	Description
Pulse Height SSD	8	0 - 255	Internal pulse height for SSD
Pulse Height Anode	8	0 - 255	Internal pulse height for anode
AB PH Thresh.	8	0 - 255	A and B pulse height threshold
CD PH Thresh.	8	0 - 255	C and D pulse height threshold
Reserved	14	0	
HV Enb	1	0 = Disable 1 = Enable	HV power supply enable
BV Enb	1	0 = Disable 1 = Enable	Bias voltage power supply enable
Comm. HV V	8	0 - 255	Common HV voltage DAC
BV V	8	0 - 255	SSD bias voltage DAC
A HV V	8	0 - 255	HV voltages DAC
B HV V	8	0 - 255	
C HV V	8	0 - 255	
D HV V	8	0 - 255	
A HV I Lim.	8	0 - 255	HV current limits DAC
B HV I Lim.	8	0 - 255	
C HV I Lim.	8	0 - 255	
D HV I Lim.	8	0 - 255	
Reserved	7	0	
HV Plug Enb	1	0 = Disable 1 = Enable	HV safing plug (digital)
A HV Trip	1	0 = Safe 1 = Over-current	HV over-current trip status
B HV Trip	1	0 = Safe 1 = Over-current	
C HV Trip	1	0 = Safe 1 = Over-current	
D HV Trip	1	0 = Safe 1 = Over-current	
A HV Fault	1	0 = Okay 1 = Fault	HV over-current monitor fault status
B HV Fault	1	0 = Okay 1 = Fault	

Name	Length (bits)	Value	Description
C HV Fault	1	0 = Okay 1 = Fault	
D HV Fault	1	0 = Okay 1 = Fault	
Reserved	12	0	
A HV Monitor	1	0 = Disable 1 = Enable	HV over-current monitor enables
B HV Monitor	1	0 = Disable 1 = Enable	
C HV Monitor	1	0 = Disable 1 = Enable	
D HV Monitor	1	0 = Disable 1 = Enable	
A Start1 CFD Thresh	16	0 - 511	A CFD thresholds
A Start2 CFD Thresh	16	0 - 511	
A Stop CFD Thresh	16	0 - 511	
B Start1 CFD Thresh	16	0 - 511	B CFD thresholds
B Start2 CFD Thresh	16	0 - 511	
B Stop CFD Thresh	16	0 - 511	
C Start1 CFD Thresh	16	0 - 511	C CFD thresholds
C Start2 CFD Thresh	16	0 - 511	
C Stop CFD Thresh	16	0 - 511	
D Start1 CFD Thresh	16	0 - 511	D CFD thresholds
D Start2 CFD Thresh	16	0 - 511	
D Stop CFD Thresh	16	0 - 511	
Ion A1 Egy. Thresh	8	0 - 255	Ion A energy thresholds
Ion A2 Egy.	8	0 - 255	

Name	Length (bits)	Value	Description
Thresh			
Ion B1 Egy. Thresh	8	0 - 255	Ion B energy thresholds
Ion B2 Egy. Thresh	8	0 - 255	
Ion C1 Egy. Thresh	8	0 - 255	Ion C energy thresholds
Ion C2 Egy. Thresh	8	0 - 255	
Ion D1 Egy. Thresh	8	0 - 255	Ion D energy thresholds
Ion D2 Egy. Thresh	8	0 - 255	
Part. A1 Egy. Thresh	8	0 - 255	Particle A energy thresholds
Part. A2 Egy. Thresh	8	0 - 255	
Part. B1 Egy. Thresh	8	0 - 255	Particle B energy thresholds
Part. B2 Egy. Thresh	8	0 - 255	
Part. C1 Egy. Thresh	8	0 - 255	Particle C energy thresholds
Part. C2 Egy. Thresh	8	0 - 255	
Part. D1 Egy. Thresh	8	0 - 255	Particle D energy thresholds
Part. D2 Egy. Thresh	8	0 - 255	
Ion A1 Ext. Egy. Thresh	8	0 - 255	Ion A extended energy thresholds
Ion A2 Ext. Egy. Thresh	8	0 - 255	
Ion B1 Ext. Egy. Thresh	8	0 - 255	Ion B extended energy thresholds
Ion B2 Ext. Egy. Thresh	8	0 - 255	
Ion C1 Ext. Egy. Thresh	8	0 - 255	Ion C extended energy thresholds
Ion C2 Ext.	8	0 - 255	

Name	Length (bits)	Value	Description
Egy. Thresh			
Ion D1 Ext. Egy. Thresh	8	0 - 255	Ion D extended energy thresholds
Ion D2 Ext. Egy. Thresh	8	0 - 255	
Part. A1 Ext. Egy. Thresh	8	0 - 255	Particle A extended energy thresholds
Part. A2 Ext. Egy. Thresh	8	0 - 255	
Part. B1 Ext. Egy. Thresh	8	0 - 255	Particle B extended energy thresholds
Part. B2 Ext. Egy. Thresh	8	0 - 255	
Part. C1 Ext. Egy. Thresh	8	0 - 255	Particle C extended energy thresholds
Part. C2 Ext. Egy. Thresh	8	0 - 255	
Part. D1 Ext. Egy. Thresh	8	0 - 255	Particle D extended energy thresholds
Part. D2 Ext. Egy. Thresh	8	0 - 255	
Reserved	5	0	
Baseline Enable	1	0 = Disable 1 = Enable	Energy baseline measurement
Baseline Channel	3	0 = A1 1 = A2 2 = B1 3 = B2 4 = C1 5 = C2 6 = D1 7 = D2	
TDC Pos. Enable	1	0 = Optional 1 = Required	Valid position calculation required enable
TDC Check Enable	1	0 = Disable 1 = Enable	TDC internal consistency check enable
TDC Only	3	0 = Disable	TDC only decimation enable

Name	Length (bits)	Value	Description
		1 = Enable all 2 = Enable 1 of 2 3 = Enable 1 of 4 4 = Enable 1 of 8 5 = Enable 1 of 16 6 = Enable 1 of 32 7 = Enable 1 of 64	
TDC Match Enable	1	0 = Disable 1 = Enable	Position calculated/SSD channel match check enable
Multi-hit Reject	1	0 = Accept 1 = Reject	Energy multiple hit reject
Reserved	2	0	
Pulse Rate	4	0 = ~300 Hz 1 = 400 Hz 2 = 500 Hz 3 = ~600 Hz 4 = ~700 Hz 5 = 800 Hz 6 = ~900 Hz 7 = 1 kHz 8 = 2 kHz 9 = ~3 kHz 10 = 4 kHz 11 = 5 kHz 12 = ~6 kHz 13 = ~7 kHz 14 = 10 kHz 15 = 25 kHz	Internal pulse rate
A Start Pulse Enb	1	0 = Disable 1 = Enable	Start and stop pulse routing
A Stop Pulse Enb	1	0 = Disable 1 = Enable	
B Start Pulse Enb	1	0 = Disable 1 = Enable	
B Stop Pulse Enb	1	0 = Disable 1 = Enable	
C Start Pulse Enb	1	0 = Disable 1 = Enable	
C Stop Pulse Enb	1	0 = Disable 1 = Enable	
D Start	1	0 = Disable	

Name	Length (bits)	Value	Description
Pulse Enb		1 = Enable	
D Stop Pulse Enb	1	0 = Disable 1 = Enable	
SSD1 Pulse Enb	1	0 = Disable 1 = Enable	SSD pulse routing
SSD2 Pulse Enb	1	0 = Disable 1 = Enable	
Pulse Start to Stop Delay	4	Unsigned integer	Pulse delays from Start to Stop, SSD1, and SSD2 (12.5ns)
Pulse Start to SSD1 Delay	6	Unsigned integer	
Pulse Start to SSD2 Delay	6	Unsigned integer	

Table 34. Software Status

Name	Length (bits)	Value	Description
Status Int.	16	1 – 65535 (0 = Off)	Status interval (seconds)
Macro Blocks	16	Unsigned integer	Number of macro blocks free
Tlm. Vol.	16	Unsigned integer	Telemetry volume produced (KiB)
Watch Addr.	16	Unsigned integer	Memory watch address
Watch Mem	8	Unsigned integer	Memory watch id (page no.)
Watch Data	2 * 8	Unsigned integer	Watched memory
Sw. Version	8	Unsigned integer	Software version number
Alarm Id	8	See Table 72	Latest alarm Id
Alarm Type	1	0 = Persistent 1 = Transient	Latest alarm type
Alarm Count	7	Unsigned integer	Count of alarms
Cmd Exec	8	Unsigned integer	Commands executed
Cmd Reject	8	Unsigned integer	Commands rejected
Mac Exec	8	Unsigned integer	Macro commands executed
Mac Reject	8	Unsigned integer	Macro commands rejected
Macro Id	8	Unsigned integer	Id of most recent macro executed
Macro Learn	1	0 = Not learning 1 = Learning	Macro learn mode
Monitor Response	1	0 = Disable 1 = Enable	Monitor response
Write Enb	1	0 = Disable 1 = Enable	Memory write enable
Reserved	4	0	
Op State	1	0 = Manual 1 = Auto	Operations state
Safing Time	16	0 – 65535	Time-out interval (seconds)
Slot 0	2	0 = Ion composition 1 = Ion energy 2 = Particle comp. 3 = Particle energy	Data collection pattern
Slot 1	2		
Slot 2	2		
Slot 3	2		
Slot 4	2		
Slot 5	2		
Slot 6	2		
Slot 7	2		
Reserved	7	0	Ion composition data product enables
Ion TOF	1	0 = Disable	

Name	Length (bits)	Value	Description
		1 = Enable	
Mass Histograms	1	0 = Disable 1 = Enable	
PH Scan	1	0 = Disable 1 = Enable	
Ions2	1	0 = Disable 1 = Enable	
Ions1	1	0 = Disable 1 = Enable	
High-Time Res. Protons	1	0 = Disable 1 = Enable	
High-Res. Protons	1	0 = Disable 1 = Enable	
Ion Comp. Diagnostic	1	0 = Disable 1 = Enable	
Ion Comp. Basic	1	0 = Disable 1 = Enable	
Reserved	5	0	Ion energy data product enables
Ion Energy	1	0 = Disable 1 = Enable	
Ion Energy Diagnostic	1	0 = Disable 1 = Enable	
Ion Energy Basic	1	0 = Disable 1 = Enable	
Reserved	4	0	Particle composition data product enables
Particle TOF	1	0 = Disable 1 = Enable	
Particles	1	0 = Disable 1 = Enable	
Particle Comp. Diagnostic	1	0 = Disable 1 = Enable	
Particle Comp. Basic	1	0 = Disable 1 = Enable	
Reserved	2	0	Particle energy data product enables
High Energy Particles	1	0 = Disable 1 = Enable	
High-Look Res. Electrons	1	0 = Disable 1 = Enable	

Name	Length (bits)	Value	Description
High-Time Res. Electrons	1	0 = Disable 1 = Enable	
High-Res. Electrons	1	0 = Disable 1 = Enable	
Particle Energy Diagnostic	1	0 = Disable 1 = Enable	
Particle Energy Basic	1	0 = Disable 1 = Enable	
Integration Time	8	1 - 255	Integration time, T (seconds)
Integration Mult. N1	8	1 - 255	Integration multiplier, N1
Integration Mult. N2	8	1 - 255	Integration multiplier, N2
Raw Ion Comp. Events	8	0 - 100 TBD	Maximum number of raw ion composition events
Raw Ion Energy Events	8	0 - 100 TBD	Maximum number of raw ion energy events
Raw Particle Comp. Events	8	0 - 100 TBD	Maximum number of raw particle composition events
Raw Particle Energy Events	8	0 - 100 TBD	Maximum number of raw particle energy events
Priority Ion Comp. Events	8	0 - 100 TBD	Maximum number of priority ion composition events
Priority Particle Comp. Events	8	0 - 100 TBD	Maximum number of priority particle composition events
Priority Particle Energy Events	8	0 - 100 TBD	Maximum number of priority particle energy events
Op Macro	8	Unsigned integer	Id of most recently executed

Name	Length (bits)	Value	Description
			operations macro
BV Limit	8	0 - 255	SSD bias voltage limit
BV Goal	8	0 - 255	SSD bias voltage goal
Comm. HV Limit	8	0 - 255	Common HV limit
Comm. HV Goal	8	0 - 255	Common HV goal
A HV Limit	8	0 - 255	HV limits
B HV Limit	8	0 - 255	
C HV Limit	8	0 - 255	
D HV Limit	8	0 - 255	
A HV Goal	8	0 - 255	HV goals
B HV Goal	8	0 - 255	
C HV Goal	8	0 - 255	
D HV Goal	8	0 - 255	
Comm. HV Level	8	0 - 255	Common HV level (intermediate to DAC)
Reserved	8	0	

8.3.6 Macro Dump

Each command saved in a macro is stored with a length, opcode, and arguments. The format is shown below. Each macro consists of a series of one or more stored commands. The last command will be EPILO_MAC_END.

Table 35. Internal Macro Command Format

Name	Length (bits)	Value	Description
Length	8	Unsigned integer	Length of remaining fields (bytes)
Reserved	8	0	
Opcode	8	Unsigned integer	Command opcode
Arguments	N * 8		Command arguments

Macro dump packets contain 32 bytes of stored macro data. Each dump packet consists of a header and macro data. The grouping flags in the header will be set to "none". The time tag will be set to the time at which the dump packet was generated. The data consists of the id of the macro being dumped, an index number of the macro block, and 32 bytes of data from the stored macro. If the final macro block is not full, it will be padded with zeros. Note: undefined macros are skipped.

Table 36. Macro Dump Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	37	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Macro Id	8	0 – 255	Id of dumped macro
Block	8	Unsigned integer	Block number (modulo 256)
Data	32 * 8		Macro data

8.3.7 Macro Checksum

Macro checksum packets report the checksums for a selected group of macros. Each checksum packet consists of a header and checksum data. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the checksums were computed. Each checksum packet identifies the range of macros checked followed by the actual checksums. Each checksum is the 16-bit CRC-CCITT of the indicated stored macro. The checksum covers the entire stored macro including possible padding in the final block (see above). Undefined macros will have a checksum of 0xffff. This is a variable-length packet.

Table 37. Macro Checksum Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	7 - 517	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
First Macro	8	0 – 255	Id of first checksummed macro
Last Macro	8	0 – 255	Id of last checksummed macro
Checksum 1	16	Unsigned integer	Computed checksum 1
Checksum 2	16	Unsigned integer	Computed checksum 2
…			

8.3.8 Data Structures

The EPI-Lo software makes several of its internal data structures available for modification or downlink. These are monitor limits and parameters. The data structures can be modified by command (EPILO_MEM_STR_LOAD). The current data structures can also be downlinked by command (EPILO_MEM_STR_READ). The packet formats are shown below. The actual data structures are described in Appendix 1.

Table 38. Monitor Limits Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	TBD	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Limits	TBD * 8	See Table 70	Monitor limits

Table 39. Parameter Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	TBD	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Parameters	TBD * 8	See Table 71	Parameter data

8.3.9 Text

Text received in a EPILO_CMD_TEXT command is sent in a packet. Each text packet consists of a header and the text. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the command was executed. This is a variable-length packet.

Table 40. Text Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5 - 81	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Text Length	8	0 - 77 (C)	Text string length
Text	C * 8		Text string
Pad	0 or 8	0	Pad up to 2-byte boundary

8.3.10 Critical Housekeeping

EPI-Lo generates a critical housekeeping packet every second. Each packet consists of a header, power requests, shared data, and housekeeping data. The grouping flags in the header will be set to "none" and the time tag will be set to the time at which the packet was generated. The eight bytes of housekeeping will be included in the spacecraft's combined instrument critical housekeeping packet. The four bytes of shared data are distributed to all SPP instruments; EPI-Lo currently sets this data to all zeros TBD. The packet format is shown in the following table.

Table 41. Critical Housekeeping Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	17	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Spare	6	0	
Power Request	2	00 = No request 01 = Power cycle 1x = Power down	Power down or power cycle request (note: power down has precedence)
Spare	8	0	
EPI-Lo Shared Data	4 * 8	0	TBD/Reserved for shared data
Status	4	TBD	Key status flags
Sensor Temp.	4	Unsigned integer	Sensor temperature
Logic Board Temp.	4	Unsigned integer	Logic board temperature
Power Board Temp.	4	Unsigned integer	Power board temperature
HVPS Current	8	Unsigned integer	High voltage power supply current
HVPS Voltage	8	Unsigned integer	High voltage power supply voltage
LVPS Current	8	Unsigned integer	Low voltage power supply current
LVPS Voltage	8	Unsigned integer	Low voltage power supply voltage
Total MCP Rate	4	Unsigned integer	Total MCP singles count rate
Total SSD Rate	4	Unsigned integer	Total SSD singles count rate
Proc. Ion Event Rate	4	Unsigned integer	Processed ion event rate

Name	Length (bits)	Value	Description
Proc. Electron Event Rate	4	Unsigned integer	Processed electron event rate

EPI-Lo can request a power down or power cycle via the power request bits in this packet. A power down can be requested by command (EPILO_SAF_OFF). Similarly, a power cycle can be requested by command (EPILO_SAF_CYCLE).

8.4 Science Packets

All science packets contain a science subheader immediately following the CCSDS secondary header. The commanded integration time is included; an abort flag is set if the integration was aborted. The following table shows the format of the science subheader.

Table 42. Science Subheader

Name	Length (bits)	Value	Description
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)

The following sections describe the EPI-Lo science packets. A number of EPI-Lo science packets exceed the maximum packet length. These are split into multiple packets; ground software uses the grouping flags to reassemble them. Such packets will be noted below; the packet length field will reflect the reassembled, i.e. original length.

8.4.1 Ion Composition

EPI-Lo produces ion composition rate counters, binned event data, and single event data science packets. These are discussed below.

8.4.1.1 Rate Counters

Basic rates packets contain basic event counts; diagnostic rates packets contain additional counts. Each packet is enabled or disabled, and its integration time is controlled by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. The following tables show the format of the basic rates and the diagnostic rates packets. The counts are log-compressed.

Table 43. Ion Composition Basic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	41	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	27 * 10	10-bit log-compressed values	Counters (see Table 6)
Pad	2	0	Zero pad to 2-byte boundary

Table 44. Ion Composition Diagnostic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	47	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	32 * 10	10-bit log-compressed values	Counters (see Table 7)
Pad	0	0	Zero pad to 2-byte boundary

8.4.1.2 Binned Event Data

Binned event data includes ion composition, ion TOF histogram, and TBD packets. Each packet is enabled or disabled, and its integration time is controlled by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration.

The following table shows the format of the high-resolution proton packet. The counts are log and fast-compressed. This is a large packet, and depending on compression, may be split into multiple packets. The packet length field reflects the reassembled, i.e. original length. This is a variable-length packet.

Table 45. High-Resolution Protons Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	167 - 6567	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 64 * var	Log and fast-compressed values	High-resolution proton counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

The following table shows the format of the high time-resolution proton packet. The counts are log and fast-compressed. This is a variable-length packet.

Table 46. High Time-Resolution Protons Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	17 - 417	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	20 * 16 * var	Log and fast-compressed values	High time-resolution proton counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

The following tables show the format of the ion packets. The counts are log and fast-compressed. These are large packets, and depending on compression, may be split into multiple packets. The packet length field reflects the reassembled, i.e. original length. These are variable-length packets.

Table 47. Ions Group1 Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	327 - 13127	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 128 * var	Log and fast-compressed values	Ion group1 counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

Table 48. Ions Group2 Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	407 - 16407	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 160 * var	Log and fast-compressed values	Ion group2 counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

TBD others, e.g. PH threshold scan

The following table shows the format of the ion TOF histogram packet. The counts are log and fast-compressed. This is a large packet, and depending on compression, may be split into multiple packets. The packet length field reflects the reassembled, i.e. original length. This is a variable-length packet.

Table 49. Ion TOF Histogram Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	167 - 6567	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 64 * var	Log and fast-compressed values	Ion TOF histogram
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.1.3 Single Event Data

Raw event packets contain raw particle event data. The number of events in the packet and its integration time are controlled by commands (EPILO_DAT_RAW and EPILO_DAT_TIME). Each packet consists of a header and raw event data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. In addition to the raw data, there is a bin number computed by the software. The following table shows the format of the raw event packet, which includes all the data. This is a variable-length packet.

Table 50. Ion Composition Raw Event Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9 - 1333	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Count	8	0 - 100	Number of events (N) in the following fields
SSD Coin Flags 1	5	See Table 2	Raw events; note: interpretation of Bin depends on Calc. Pos. and SSD Chan
MCP Coin Flags 1	2	See Table 2	
TDC Valid Flags 1	3	See Table 2	
Start1 Stop TDC 1	11	0 - 2047	
Start1 Start2 TDC 1	11	0 - 2047	
Start2 Stop TDC 1	11	0 - 2047	
SSD Energy 1	12	0 - 4095	
SSD PW 1	11	0 - 2047	
Ext. Energy PW 1	11	0 - 2047	
Calc. TOF 1	11	0 - 2047	
Quadrant 1	2	See Table 2	
SSD Chan 1	2	See Table 2	
Calc. Pos. 1	5	See Table 2	

Name	Length (bits)	Value	Description
Bin 1	9	0 - 351 (H+/ions) 0 - 63 (Ion TOF)	
...			
SSD Coin Flags N	5	See Table 2	
MCP Coin Flags N	2	See Table 2	
TDC Valid Flags N	3	See Table 2	
Start1 Stop TDC N	11	0 - 2047	
Start1 Start2 TDC N	11	0 - 2047	
Start2 Stop TDC N	11	0 - 2047	
SSD Energy N	12	0 - 4095	
SSD PW N	11	0 - 2047	
Ext. Energy PW N	11	0 - 2047	
Calc. TOF N	11	0 - 2047	
Quadrant N	2	See Table 2	
SSD Chan N	2	See Table 2	
Calc. Pos. N	5	See Table 2	
Bin N	9	0 - 351 (H+/ions) 0 - 63 (Ion TOF)	
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.2 Ion Energy

EPI-Lo produces ion energy rate counters, binned event data, and single event data science packets. These are discussed below.

8.4.2.1 Rate Counters

Basic rates packets contain basic event counts; diagnostic rates packets contain additional counts. Each packet is enabled or disabled, and its integration time is controlled, by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. The following table shows the format of the basic rates and the diagnostic rates packets. The counts are log-compressed.

Table 51. Ion Energy Basic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	41	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	27 * 10	10-bit log-compressed values	Counters (see Table 8)
Pad	2	0	Zero pad to 2-byte boundary

Table 52. Ion Energy Diagnostic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	23	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	12 * 10	10-bit log-compressed values	Counters (see Table 9)
Pad	8	0	Zero pad to 2-byte boundary

8.4.2.2 Binned Event Data

Binned event data of an ion energy histogram packet. The packet is enabled or disabled, and its integration time is controlled by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). The packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. The following table shows the format of the ion energy histogram packet. The counts are log and fast-compressed. This is a variable-length packet.

Table 53. Ion Energy Histogram Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	23 - 663	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	8 * 64 * var	Log and fast-compressed values	Ion energy histogram
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.2.3 Single Event Data

Raw event packets contain raw particle event data. The number of events in the packet and its integration time are controlled by commands (EPILO_DAT_RAW and EPILO_DAT_TIME). Each packet consists of a header and raw event data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. In addition to the raw data, there is a bin number computed by the software. The following table shows the format of the raw event packet, which includes all the data. This is a variable-length packet.

Table 54. Ion Energy Raw Event Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9 - 1297	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Count	8	0 - 100	Number of events (N) in the following fields
SSD Coin Flags 1	5	See Table 2	Raw events; note: interpretation of Bin depends on Calc. Pos.
MCP Coin Flags 1	2	See Table 2	
TDC Valid Flags 1	3	See Table 2	
Start1 Stop TDC 1	11	0 - 2047	
Start1 Start2 TDC 1	11	0 - 2047	
Start2 Stop TDC 1	11	0 - 2047	
SSD Energy 1	12	0 - 4095	
SSD PW 1	11	0 - 2047	
Ext. Energy PW 1	11	0 - 2047	
Calc. TOF 1	11	0 - 2047	
Quadrant 1	2	See Table 2	
SSD Chan 1	2	See Table 2	
Calc. Pos. 1	5	See Table 2	

Name	Length (bits)	Value	Description
Bin 1	6	0 - 63	
...			
SSD Coin Flags N	5	See Table 2	
MCP Coin Flags N	2	See Table 2	
TDC Valid Flags N	3	See Table 2	
Start1 Stop TDC N	11	0 - 2047	
Start1 Start2 TDC N	11	0 - 2047	
Start2 Stop TDC N	11	0 - 2047	
SSD Energy N	12	0 - 4095	
SSD PW N	11	0 - 2047	
Ext. Energy PW N	11	0 - 2047	
Calc. TOF N	11	0 - 2047	
Quadrant N	2	See Table 2	
SSD Chan N	2	See Table 2	
Calc. Pos. N	5	See Table 2	
Bin N	6	0 - 63	
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.3 Particle Composition

EPI-Lo produces particle composition rate counters, binned event data, and single event data science packets. These are discussed below.

8.4.3.1 Rate Counters

Basic rates packets contain basic event counts; diagnostic rates packets contain additional counts. Each packet is enabled or disabled, and its integration time is controlled, by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. The following table shows the format of the basic rates and the diagnostic rates packets. The counts are log-compressed.

Table 55. Particle Composition Basic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	51	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	35 * 10	10-bit log-compressed values	Counters (see Table 10)
Pad	2	0	Zero pad to 2-byte boundary

Table 56. Particle Composition Diagnostic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	27	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	16 * 10	10-bit log-compressed values	Counters (see Table 11)
Pad	0	0	Zero pad to 2-byte boundary

8.4.3.2 Binned Event Data

Binned event data includes particle composition and particle TOF histogram. Each packet is enabled or disabled, and its integration time is controlled by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration.

The following table shows the format of the particle composition packet. The counts are log and fast-compressed. This is a large packet, and depending on compression, may be split into multiple packets. The packet length field reflects the reassembled, i.e. original length. This is a variable-length packet. TBD: this packet may need to be split.

Table 57. Particle Composition Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	887 - 36087	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 352 * var	Log and fast-compressed values	Particle counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

The following table shows the format of the particle TOF histogram packet. The counts are log and fast-compressed. This is a large packet, and depending on compression, may be split into multiple packets. The packet length field reflects the reassembled, i.e. original length. This is a variable-length packet.

Table 58. Particle TOF Histogram Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	00 = Continuation 01 = First packet 10 = Last packet 11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	167 - 6567	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 64 * var	Log and fast-compressed values	Particle TOF histogram
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.3.3 Single Event Data

Raw event packets contain raw particle event data. The number of events in the packet and its integration time are controlled by commands (EPILO_DAT_RAW and EPILO_DAT_TIME). Each packet consists of a header and raw event data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. In addition to the raw data, there is a bin number computed by the software. The following table shows the format of the raw event packet, which includes all the data. This is a variable-length packet.

Table 59. Particle Composition Raw Event Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9 - 1333	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Count	8	0 - 100	Number of events (N) in the following fields
SSD Coin Flags 1	5	See Table 2	Raw events; note: interpretation of Bin depends on Calc. Pos. and SSD Chan
MCP Coin Flags 1	2	See Table 2	
TDC Valid Flags 1	3	See Table 2	
Start1 Stop TDC 1	11	0 - 2047	
Start1 Start2 TDC 1	11	0 - 2047	
Start2 Stop TDC 1	11	0 - 2047	
SSD Energy 1	12	0 - 4095	
SSD PW 1	11	0 - 2047	
Ext. Energy PW 1	11	0 - 2047	
Calc. TOF 1	11	0 - 2047	
Quadrant 1	2	See Table 2	
SSD Chan 1	2	See Table 2	
Calc. Pos. 1	5	See Table 2	

Name	Length (bits)	Value	Description
Bin 1	9	0 - 351 (Particles) 0 - 63 (Part. TOF)	
...			
SSD Coin Flags N	5	See Table 2	
MCP Coin Flags N	2	See Table 2	
TDC Valid Flags N	3	See Table 2	
Start1 Stop TDC N	11	0 - 2047	
Start1 Start2 TDC N	11	0 - 2047	
Start2 Stop TDC N	11	0 - 2047	
SSD Energy N	12	0 - 4095	
SSD PW N	11	0 - 2047	
Ext. Energy PW N	11	0 - 2047	
Calc. TOF N	11	0 - 2047	
Quadrant N	2	See Table 2	
SSD Chan N	2	See Table 2	
Calc. Pos. N	5	See Table 2	
Bin N	9	0 - 351 (Particles) 0 - 63 (Part. TOF)	
Pad	0 - 15	0	Zero pad to 2-byte boundary

8.4.4 Particle Energy

EPI-Lo produces particle energy rate counters, binned event data, and single event data science packets. These are discussed below.

8.4.4.1 Rate Counters

Basic rates packets contain basic event counts; diagnostic rates packets contain additional counts. Each packet is enabled or disabled, and its integration time is controlled, by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. The following table shows the format of the basic rates and the diagnostic rates packets. The counts are log-compressed.

Table 60. Particle Energy Basic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	51	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	35 * 10	10-bit log-compressed values	Counters (see Table 12)
Pad	2	0	Zero pad to 2-byte boundary

Table 61. Particle Energy Diagnostic Rates Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	17	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	8 * 10	10-bit log-compressed values	Counters (see Table 13)
Pad	0	0	Zero pad to 2-byte boundary

8.4.4.2 Binned Event Data

Binned event data includes electrons, and TBD packets. Each packet is enabled or disabled, and its integration time is controlled by commands (EPILO_DAT_ENB and EPILO_DAT_TIME). Each packet consists of a header and science data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration.

The following table shows the format of the high-resolution electron packet. The counts are log and fast-compressed. This is a variable-length packet.

Table 62. High-Resolution Electrons Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	23 - 663	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	8 * 64 * var	Log and fast-compressed values	High-resolution electron counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

The following table shows the format of the high time-resolution electron packet. The counts are log and fast-compressed. This is a variable-length packet.

Table 63. High Time-Resolution Electrons Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	11 - 171	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	8 * 16 * var	Log and fast-compressed values	High time-resolution electron counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

The following table shows the format of the high look-resolution electron packet. The counts are log and fast-compressed. This is a variable-length packet.

Table 64. High Look-Resolution Electrons Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	87 - 3287	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Counts	80 * 32 * var	Log and fast-compressed values	High-resolution proton counts
Pad	0 - 15	0	Zero pad to 2-byte boundary

TBD: missing high-energy particles

8.4.4.3 Single Event Data

Raw event packets contain raw particle event data. The number of events in the packet and its integration time are controlled by commands (EPILO_DAT_RAW and EPILO_DAT_TIME). Each packet consists of a header and raw event data. There is a science subheader which includes the integration time. The time-tag indicates the start of the integration. In addition to the raw data, there is a bin number computed by the software. The following table shows the format of the raw event packet, which includes all the data. This is a variable-length packet.

Table 65. Particle Energy Raw Event Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	9 - 1297	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Reserved	7	0	
Abort Flag	1	0 = Ok 1 = Aborted	Integration aborted?
Reserved	8	0	
Int. Time	16	Unsigned integer	Integration time (seconds)
Count	8	0 - 100	Number of events (N) in the following fields
SSD Coin Flags 1	5	See Table 2	Raw events; note: interpretation of Bin depends on Calc. Pos.
MCP Coin Flags 1	2	See Table 2	
TDC Valid Flags 1	3	See Table 2	
Start1 Stop TDC 1	11	0 - 2047	
Start1 Start2 TDC 1	11	0 - 2047	
Start2 Stop TDC 1	11	0 - 2047	
SSD Energy 1	12	0 - 4095	
SSD PW 1	11	0 - 2047	
Ext. Energy PW 1	11	0 - 2047	
Calc. TOF 1	11	0 - 2047	
Quadrant 1	2	See Table 2	
SSD Chan 1	2	See Table 2	
Calc. Pos. 1	5	See Table 2	

Name	Length (bits)	Value	Description
Bin 1	6	0 - 63	
...			
SSD Coin Flags N	5	See Table 2	
MCP Coin Flags N	2	See Table 2	
TDC Valid Flags N	3	See Table 2	
Start1 Stop TDC N	11	0 - 2047	
Start1 Start2 TDC N	11	0 - 2047	
Start2 Stop TDC N	11	0 - 2047	
SSD Energy N	12	0 - 4095	
SSD PW N	11	0 - 2047	
Ext. Energy PW N	11	0 - 2047	
Calc. TOF N	11	0 - 2047	
Quadrant N	2	See Table 2	
SSD Chan N	2	See Table 2	
Calc. Pos. N	5	See Table 2	
Bin N	6	0 - 63	
Pad	0 - 15	0	Zero pad to 2-byte boundary

9. Boot Program

The boot program for EPI-Lo includes many of the features described above, but excludes others. The boot program does not provide macro services, monitoring, or autonomy/safing. The boot program does support the telemetry and command subsets described below.

9.1 Start-Up

The boot ROM program tests RAM searching for a place to load itself. It tests pages 7 down through page 1. The boot program copies itself into the first RAM page that tests good; this will normally be page 7. No telemetry appears and no commands are accepted until after the RAM test.

Once in RAM, the boot program determines the reset cause, i.e. either power-on or watchdog. If a watchdog reset occurred, a 64 KiB memory dump starts from address 0. The reset cause is reported in the boot status (see below). Finally, all of the discriminator thresholds are set to their maximum values TBD.

EPI-Lo must be able to operate autonomously, i.e. independently of ground or spacecraft commands. This includes automatically booting the EPI-Lo application program. The boot program examines a start-up flag in the time and status message received from the spacecraft. If the flag indicates that autonomous operation is required, then the boot program attempts to boot an EPI-Lo application program stored in non-volatile memory. It starts with the application program stored in the default location (address = 0x400000). If this fails, e.g. because of a bad checksum, a second location is tried, 64KiB beyond the default. If this also fails, a third location is tried, another 64KiB beyond. If all three locations fail, an alarm is generated; no further attempts to boot are made.

9.2 Saved Programs

The boot program can load and run programs saved in non-volatile memory. This is only done in response to a command (EPILO_ROM_BOOT or EPILO_ROM_GO). The save program format consists of a program image prefixed with a five word header:

page	address	start address	length	checksum

The page and address identifies the destination for the program. The start address is the entry point of the program. The length is the total length including the header (measured in 16-bit words). The checksum is the negation of the word-wise two's complement sum of the rest of the header and the program; therefore the checksum of the header and program should be zero. If the checksum is good and the destination page is not the same as the page the boot program is using, the program image is copied to RAM; after the copy, the image in RAM is checked. If all of these steps are successful, the new program is started at the given address. Otherwise, the boot program continues waiting for commands.

9.3 Commands

The boot program processes commands received from its host. The boot program accepts a subset of the application commands:

- EPILO_CMD_NULL and EPILO_CMD_WRAP
- EPILO_MEM_CHECK, EPILO_MEM_COPY, EPILO_MEM_LOAD, EPILO_MEM_READ, EPILO_MEM_READ_ABT, EPILO_MEM_RUN, and EPILO_MEM_WR_EN
- EPILO_STAT_CLR, EPILO_STAT_INT, and EPILO_STAT_MEM

The program also accepts some boot-specific commands.

9.3.1 EPILO_ROM_BOOT – Boot Default Program

Boot the default program from non-volatile memory. See EPILO_ROM_GO below.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	1	Packet length (bytes) - 1
Opcode	8	0x1e	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control

9.3.2 EPILO_ROM_GO – Boot a Selected Program

Boot a program. The address identifies a boot header (described above). The command will be rejected if the program has a bad checksum.

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	1	Designates a telecommand packet
Secondary?	1	0	Secondary header is absent
APID	11	0x4a0	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	5	Packet length (bytes) - 1
Opcode	8	0x1f	Opcode
Macro?	8	0 = Execute 1 = Append to macro	Macro learn control
Address	32	Unsigned integer	Program header address

9.4 Telemetry

The boot program generates a subset of the telemetry generated by the flight program. Memory dump packets and memory checksum packets can be requested. Command echoes are generated. A short boot status packet is generated periodically. The rate is controlled by a command (see EPILO_STAT_INT). Each boot status packet consists of a header and status data. The grouping flags in the header will be set to “none” and the time tag will be set to the time at which the status was collected.

Table 66. Boot Status Packet

Name	Length (bits)	Value	Description
Version	3	000	Designates a source packet
Type	1	0	Designates a telemetry packet
Secondary?	1	1	Secondary header is present
APID	11	See Table 24	Application Process ID
Grouping	2	11 = None	Grouping flags
Sequence Count	14	Unsigned integer	Continuous sequence count for each Application Process ID
Length	16	19	Packet length (bytes) - 1
Secondary Header	32	Unsigned integer	Time tag (MET)
Status Int.	16	1 – 65535 (0 = Off)	Status interval (seconds)
Tlm Vol.	16	Unsigned integer	Telemetry volume produced (KiB)
Watch Addr.	16	Unsigned integer	Memory watch address
Watch Mem	8	Unsigned integer	Memory watch id (page no.)
Watch Data	2 * 8	Unsigned integer	Watched memory
Sw. Version	8	0 = Boot	Software version number
Alarm Id	8	See Table 72	Latest alarm Id
Alarm Type	1	0 = Persistent 1 = Transient	Latest alarm type
Alarm Count	7	Unsigned integer	Count of alarms
Cmd Exec	8	Unsigned integer	Commands executed
Cmd Reject	8	Unsigned integer	Commands rejected
Cause	1	0 = Normal 1 = Watchdog	Reset cause
Write Enb	1	0 = Disable 1 = Enable	Memory write enable
Reserved	3	0	
Code Ram	3	1 - 7	RAM page selected for running boot program
Reserved	8	0	

Note: the critical housekeeping packet is not generated in boot.

9.5 Serial Test Port

A serial test port is available during ground testing. Typing a carriage return any time that the boot program is running brings up an ASCII interface. A set of commands similar to that available over the spacecraft interface is provided. Each command is invoked by a single character:

Table 67. Test Port Commands

Command	Arguments	Action
b		Boot (from default location)
c	src<cr>dst<cr>len<cr>	Copy Memory
e	entry<cr>	Execute Program
g	addr<cr>	Go (boot from selected location)
i	addr<cr>	Inspect/Modify Memory
l		Load Memory (in Motorola S-record format)

10. Usage

This section contains advice on using the EPI-Lo software.

After EPI-Lo is booted, the commanded state, with a few exceptions, is set to all zeros. For example, the MCP HV limit is set to 0. The exceptions are the status interval is set to 1 (EPILO_STAT_INT), safing timeout is set to 5 seconds (EPILO_SAF_TIMEOUT), the thresholds are all set to maximum (EPILO_EGY_THRESH and EPILO_TOF_THRESH and EPILO_TOF_CFD), and the pixel rate thresholds are all set to maximum (EPILO_EGY_PIX_AUTO). TBD Puck

There are a number of constraints for using the macro system. The following rules must be observed by the user:

- Do not nest too deeply. One macro can call another or can execute loops. Both calls and loops can be nested. Loop nesting depth is traded with macro nesting depth. Each running macro context has a 32 element stack. Each macro call uses two elements; each loop uses three elements.
- EPILO_MAC_SAVE and EPILO_MAC_RESTORE must not be used while a macro is being compiled. Similarly, macro dumps via EPILO_MAC_READ and macro define cannot occur at the same time.
- EPILO_MAC_LOOP_BEG and EPILO_MAC_LOOP_END must be paired correctly. Each EPILO_MAC_LOOP_BEG must be followed by a EPILO_MAC_LOOP_END.
- Avoid macro loops without EPILO_MAC_DELAY occurring in the loop. This could shut out low priority processes and cause a watchdog reset.
- Do not use entire macro space. If there are no free macro blocks, all subsequent macro compilations will fail.

Some commands take substantial time to execute. Most commands execute in microseconds, but a few, specifically MRAM writes, take several seconds to execute. If such a command is executed, a delay should be inserted before sending the next command. Slow commands are: TBD: re-evaluate; MRAM is faster than EEPROM

- EPILO_MAC_SAVE
- EPILO_MAC_CHECK if a large number of macros are checked
- EPILO_MEM_COPY if a large block of memory is copied to MRAM
- EPILO_MEM_CHECK if a large block of memory is checked

The following table summarizes the size and location of the processor's RAM, memory-mapped I/O, MRAM, and boot PROM. The memory commands, EPILO_MEM_CHECK, EPILO_MEM_COPY, EPILO_MEM_LOAD, and EPILO_MEM_READ are fully supported given a destination address in RAM or MRAM. The PROM cannot be loaded, but can be read with EPILO_MEM_COPY or EPILO_MEM_READ and checked using EPILO_MEM_CHECK. A more complete description of the memory map appears in Reference 3.

Table 68. Memory Map

Address	Resource	Notes
00.0000 (16x64KiB)	RAM0	
10.0000 (16x64KiB)	I/O (and ROM)	See below
20.0000 (16x64KiB)	RAM0	Alias for above
30.0000 (16x64KiB)	RAM1	
40.0000 (16x64KiB)	MRAM	
60.0000 (16x64KiB)	MRAM (alias)	

The I/O section of the memory map, which also includes a ROM, is expanded in the following table.

Table 69. I/O (and ROM) Memory Map

Address	Resource	Notes
10.0000 (32KiB)	Core I/O	16-bit I/O only
10.8000 (32KiB)	Event I/O	16-bit I/O only
11.0000 (3x64KiB)	*Alias*	
14.0000 (4x64KiB)	Event	16-bit I/O only
18.0000 (4x64KiB)	*Alias*	
1C.0000 (4x64KiB)	ROM (32KiB)	16/8-bit read only

11. References

[1] APL, 7464-9004, "EPI-Lo Software Requirements Document".

[2] APL, 7434-9066, Solar Probe Plus (SPP) General Instrument (GI) ICD.

[3] Hayes, J.R., "Solar Probe Plus EPI-Lo FPGA Software Interface", TBD, SRI-TBD-TBD.

Appendix 1 – Data Structures

Monitor limits consists of lower and upper limits of each monitored value; the lower limit is first followed by the upper limit. Each limit is an eight bit unsigned integer; the monitored value is scaled accordingly before being monitored.

Table 70. Monitor Limits TBD Puck

Name	Length (bits)	Value	Description
LVPS/0	2 * 8	Unsigned integer	+1.5V voltage
LVPS/1	2 * 8	Unsigned integer	+3.3V voltage
LVPS/2	2 * 8	Unsigned integer	+5V voltage
LVPS/3	2 * 8	Unsigned integer	+15V voltage
LVPS/4	2 * 8	Unsigned integer	+15V current
LVPS/5	2 * 8	Unsigned integer	+5V current
LVPS/6	2 * 8	Unsigned integer	+3.3V current
LVPS/7	2 * 8	Unsigned integer	+1.5V current
LVPS/8	2 * 8	Unsigned integer	CFD voltage
LVPS/9	2 * 8	Unsigned integer	Primary current
LVPS/10	2 * 8	Unsigned integer	SSD bias voltage power
LVPS/11	2 * 8	Unsigned integer	SSD bias voltage
LVPS/12	2 * 8	Unsigned integer	MCP HV current
LVPS/13	2 * 8	Unsigned integer	MCP HV voltage
LVPS/14	2 * 8	Unsigned integer	LVPS temperature
LVPS/15	2 * 8	Unsigned integer	Puck temperature

Parameters are values that control the software, but are rarely or never expected to change. There are parameters for the high voltages, reference voltage DACs, etc.

Table 71. Parameters

Name	Length (bits)	Value	Description
MCP HV Clock	8	Unsigned integer	MCP HV clock rate (5MHz/(N+1))
MCP HV Ramp Up Rate	8	Unsigned integer	MCP HV ramp up rate (s/step)
MCP HV Ramp Down Rate	8	Unsigned integer	MCP HV ramp down rate (step/s)
MCP HV Safe Limit	8	Unsigned integer	MCP HV safe limit, i.e., maximum value to use if safing plug installed
SSD Bias Clock	8	Unsigned integer	SSD bias clock rate (5MHz/(N+1))
SSD Bias Ramp Up Rate	8	Unsigned integer	SSD bias ramp up rate (s/step)
SSD Bias Ramp Down Rate	8	Unsigned integer	SSD bias ramp down rate (step/s)
SSD Reference	8	Unsigned integer	SSD reference DAC setting
PH Reference	8	Unsigned integer	PH reference DAC setting
Spare	8		
AB CFD Auto-Zero	16	Bit vector	CFD auto-zero control (see Reference 4)
CD CFD Auto-Zero	16	Bit vector	
AB TDC Delay	16	Bit vector	TDC programmable delays (see Reference 4)
CD TDC Delay	16	Bit vector	
TDC Required	16	Bit vector	TDC required (see Reference 4)
A TDC Check Close	7	Unsigned integer	A TDC consistency check (see Reference 4)
A TDC Check Delay	9	Unsigned integer	
B TDC	7	Unsigned integer	B TDC consistency check (see

Name	Length (bits)	Value	Description
Check Close			Reference 4)
B TDC Check Delay	9	Unsigned integer	
C TDC Check Close	7	Unsigned integer	C TDC consistency check (see Reference 4)
C TDC Check Delay	9	Unsigned integer	
D TDC Check Close	7	Unsigned integer	D TDC consistency check (see Reference 4)
D TDC Check Delay	9	Unsigned integer	
A Position Thresholds	20 * 16	Bit vector	A position thresholds (0 - 19) and internal pulse threshold (see Reference 4)
A Int. Pulse Thresholds	16	Bit vector	
B Position Thresholds	20 * 16	Bit vector	B position thresholds (0 - 19) and internal pulse threshold (see Reference 4)
B Int. Pulse Thresholds	16	Bit vector	
C Position Thresholds	20 * 16	Bit vector	C position thresholds (0 - 19) and internal pulse threshold (see Reference 4)
C Int. Pulse Thresholds	16	Bit vector	
D Position Thresholds	20 * 16	Bit vector	D position thresholds (0 - 19) and internal pulse threshold (see Reference 4)
D Int. Pulse Thresholds	16	Bit vector	
Ion Energy Enable	8	Bit vector	Ion energy enable/disable (see Reference 4)
Particle Energy Enable	8	Bit vector	Particle energy enable/disable (see Reference 4)
Ion A1 Energy Base.	16	0 - 4095	Ion energy baselines
Ion A2 Energy	16	0 - 4095	

Name	Length (bits)	Value	Description
Base.			
Ion B1 Energy Base.	16	0 - 4095	
Ion B2 Energy Base.	16	0 - 4095	
Ion C1 Energy Base.	16	0 - 4095	
Ion C2 Energy Base.	16	0 - 4095	
Ion D1 Energy Base.	16	0 - 4095	
Ion D2 Energy Base.	16	0 - 4095	
Particle A1 Energy Base.	16	0 - 4095	Particle energy baselines
Particle A2 Energy Base.	16	0 - 4095	
Particle B1 Energy Base.	16	0 - 4095	
Particle B2 Energy Base.	16	0 - 4095	
Particle C1 Energy Base.	16	0 - 4095	
Particle C2 Energy Base.	16	0 - 4095	
Particle D1 Energy Base.	16	0 - 4095	
Particle D2	16	0 - 4095	

Name	Length (bits)	Value	Description
Energy Base.			
Ion Energy Coincidence	8	Bit vector	Ion energy coincidence definition (see Reference 4)
Particle Energy Coincidence	8	Bit vector	Particle energy coincidence definition (see Reference 4)
Particle Composition Coincidence	8	Bit vector	Particle composition coincidence definition (see Reference 4)
Spare	8	0	
Ion Comp. Basic Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	Integration cadence
Ion Comp. Diag. Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
High-Res. Protons Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
High Time-Res. Protons Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ions1 Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ions2 Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
PH Scan Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Mass Histograms	8	0 = Fast 1 = Medium1	

Name	Length (bits)	Value	Description
		2 = Medium2 3 = Slow	
Ion TOF Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Energy Basic Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Energy Diag. Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Energy Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particle Comp. Basic Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particle Comp. Diag. Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particles Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particle TOF Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particle Energy Basic Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Particle Energy Diag.	8	0 = Fast 1 = Medium1 2 = Medium2	

Name	Length (bits)	Value	Description
Cadence		3 = Slow	
High-Res. Electrons Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
High Time-Res. Electrons Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
High Look-Res Electrons Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
High-Energy Particles Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Comp. Raw Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Energy Raw Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Part. Comp. Raw Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Part. Energy Raw Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Ion Comp. Priority Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Part. comp. Priority Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	

Name	Length (bits)	Value	Description
Part. Energy Priority Cadence	8	0 = Fast 1 = Medium1 2 = Medium2 3 = Slow	
Spare	8	0	

Appendix 2 – Alarm & Result Codes

Table 72. Alarms

ID	Description	Information
0	*Unused*	
1	Bad packet	1 = ITF bad length 2 = ITF bad checksum 3 = CCSDS bad format 4 = CCSDS bad APID 5 = CCSDS bad length
2	Out of macro contexts	Macro Id
3	Autonomous operations error	0 = Bad program checksum 1 = Bad macro checksum
4 … 15	*Reserved*	
16	MCP HV over-current	0 = A 1 = B 2 = C 3 = D
7 … 127	*Reserved*	
128 … 143	Monitored value is too low	Value and limit
144 … 191	*Reserved*	
192 … 207	Monitored value is too high	Value and limit
208 … 255	*Reserved*	

Table 73. Command Results

Result Code (hex)	Description
0x00	No error, command executed
0x01	No error, command appended to macro
0x02	Unknown opcode or insufficient arguments
0x03	Bad argument
0x04	Cannot run macro; no contexts
0x05	Cannot be used outside of a macro
0x06	Macro compilation error
0x07	Macro not killed (not running?)
0x08	Cannot boot program; bad checksum
0x09	Cannot restore macros; bad checksum
0x0a	Cannot load memory; write disabled
0x0b … 0x0f	*Reserved*
0x10	HV goal greater than limit
0x11	Common/MCP HV invariant violation
0x12	All operations regions defined
0x13 … 0x7f	*Reserved*

Appendix 3 – Acronyms

Table 74. Acronyms

Acronym	Definition
1PPS	1 Pulse-Per-Second
ADC	Analog to Digital Converter
APID	Application Process Identifier
C&DH	Command & Data Handling
CCITT	International Consultative Committee on Telegraphy and Telephony
CCSDS	Consultative Committee for Space Data Systems
CFD	Constant Fraction Discriminator
CRC	Cyclic Redundancy Code
DAC	Digital to Analog Converter
EPI-Hi	Energetic Particle Instrument - High Energy
EPI-Lo	Energetic Particle Instrument - Low Energy
FIFO	First In First Out
FPGA	Field Programmable Gate Array
HV	High Voltage
ICD	Interface Control Document
I^2C	Inter-Integrated Circuit
ISIS	Integrated Science Investigation of the Sun
ITF	Instrument Transfer Frame
KiB	Kibibyte, i.e. 1024 bytes
LVPS	Low Voltage Power Supply
MCP	Micro Channel Plate
MRAM	Magnetoresistive RAM
N/A	Not Applicable
PH	Pulse Height
RAM	Random Access Memory
ROM	Read Only Memory
S/C	Spacecraft
SBC	Single Board Computer
SPI	Serial Peripheral Interface
SPP	Solar Probe Plus
SSD	Solid State Detector
SSR	Solid State Recorder
TBD	To Be Determined
TOF	Time Of Flight
UART	Universal Asynchronous Receiver Transmitter

FSCM NO.	SIZE	DRAWING NO.	REV.
88898	A	7464-9005	d

SCALE NONE	DO NOT SCALE PRINT	SHEET 181 OF 181