

Solar Probe Plus

A NASA Mission to Touch the Sun

Integrated Science Investigation of the Sun Energetic Particles

Preliminary Design Review

05 – 06 NOV 2013

EPI-Lo Software

John Hayes

EPI-Lo FSW Lead (JHU/APL)



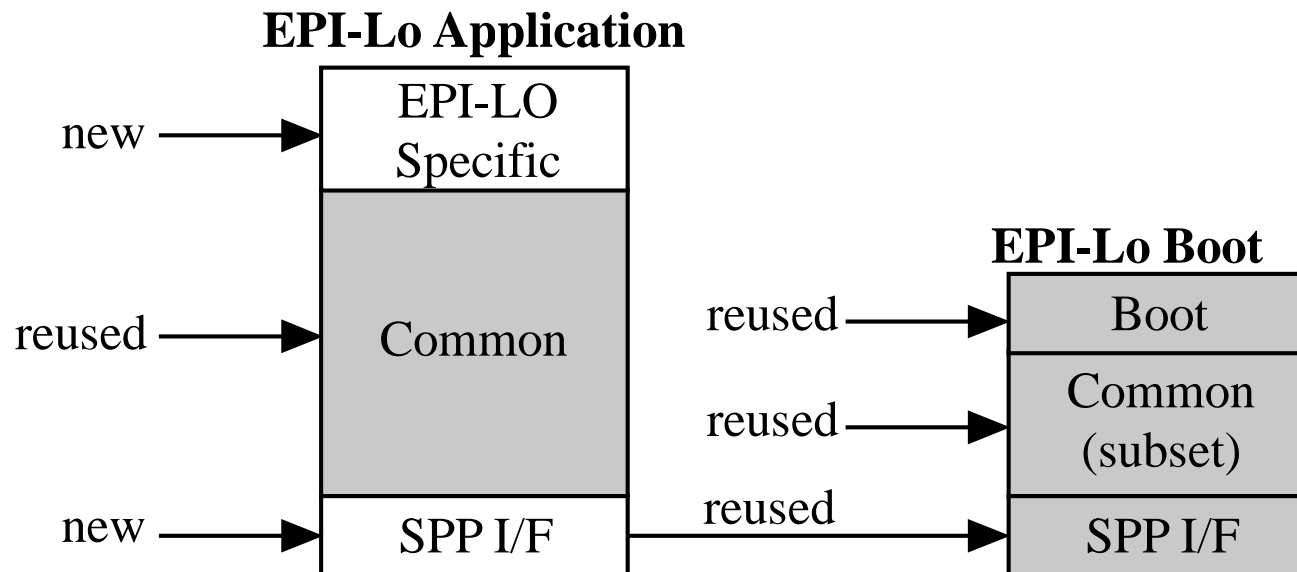
This document contains technical data that may be controlled by the International Traffic in Arms Regulations (22 CFR 120-130) and may not be provided, disclosed or transferred in any manner to any person, whether in the U.S. or abroad, who is not 1) a citizen of the United States, 2) a lawful permanent resident of the United States, or 3) a protected individual as defined by 8 USC 1324b(a)(3), without the written permission of the United States Department of State.



Outline



- Common Software
- SPP Host Software
- EPI-Lo Application Software
- EPI-Lo Boot Software
- Software Development Environment
- Summary





Common Software



- Packet telemetry
- Command handling
- Macros (stored command sequences)
- Memory management
- Monitoring and alarm generation
- Status reporting



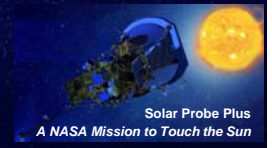
Common Software Reuse



- Common software reuse estimates for EPI-Lo
 - ~50% of application
 - >90% of boot
- Common software currently (or soon to be) in flight:
 - MESSENGER: EPPS, GRS, MAG, MASCS, MDIS, NS, and XRS
 - MRO: CRISM
 - New Horizons: LORRI and PEPSSI
 - Pucks: Juno/JEDI, RBSP/RBSPICE, MMS/EIS (2014)
 - BepiColombo/Strofi (2015)
 - Solar Orbiter/SIS (2017)
- Automated regression test for common software (and for boot software)



SPP Spacecraft Interface Software



- Command and telemetry use 115200 baud UART protocol, 8 data bits plus odd parity
- Virtual 1PPS, i.e. falling edge of start bit of first byte in command ITF
- Redundant interface, side A vs. side B
 - Command arrival determines active side; telemetry sent only to that side
 - Dynamic side switching supported
- Interrupt driven: per-byte interrupt for command and telemetry, and side A and B 1PPS interrupts



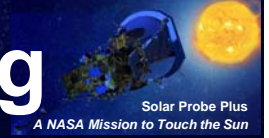
SPP Host S/W – EPI-Lo to S/C



- Telemetry ITF
 - Variable length up to 8196 bytes (EPI-Lo limits to 2060 bytes)
 - Multiple telemetry ITFs can be sent but cannot straddle 1PPS (EPI-Lo sends one per second)
 - Frame header includes sync pattern, length, etc.; checksum at end of frame
 - Frame transmission synchronized to 1PPS
- CCSDS telemetry packets
 - Variable length, up to 4096 bytes (including headers)
 - Zero to many packets per frame
 - Packets can straddle frames
 - 64 APIDs available
 - Critical housekeeping packet



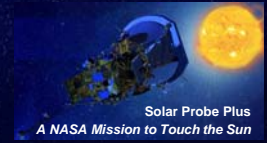
SPP Host S/W – Critical Housekeeping



- Critical housekeeping packet sent once per second
 - May optionally be sent to the SSR
 - If so, it counts against telemetry volume allocation
- Contains 32 bits of TBD data to share with other instruments
- Contains 8 bytes of housekeeping data
 - Data is included in spacecraft-generated combined instrument critical housekeeping packet
 - Data is NOT monitored by spacecraft autonomy
- Contains bits to request power off or power cycle, set via EPI-Lo command:
 - EPILO_SAF_OFF: Request power off
 - EPILO_SAF_CYCLE: Request power cycle



SPP Host S/W – S/C to EPI-Lo



- Command ITF
 - Variable length up to 512 bytes
 - S/C sends one command ITF per second
 - Frame header includes sync pattern, length, etc.; checksum at end of frame
 - Frame transmission synchronized to 1PPS
- CCSDS telecommand packets
 - Variable length, up to 362 (TBD) bytes (including headers)
 - One to many packets per frame
 - S/C time and status (telemetry) packet sent every second, always first
 - Zero or more command packets
 - Packets cannot straddle frames
 - Secondary header is optional (EPI-Lo does not use)
 - 64 APIDs available (EPI-Lo uses one)
- If command ITF is not received for a commandable number of seconds, EPI-Lo runs its safing macro



SPP Host S/W – Time Keeping



- Time is set by spacecraft
 - 32-bit MET (seconds)
 - Time arrives every second (as part of time and status)
 - Virtual 1PPS from S/C generates interrupt; interrupt routine sets time, converted to milliseconds
- Time is updated by EPI-Lo
 - 1000 Hz interrupt routine updates time
 - Common software telemetry and command processes run at 1PPS, use coarse (1 s. resolution) time tags
 - Science can be tagged with TBD finer resolution



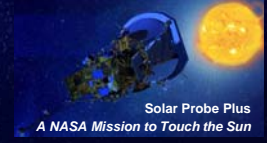
EPI-Lo Application Software



- Subsystems
 - MCP HV
 - SSD bias
 - Time-of-flight and angle/position measurement (in backup)
 - Energy measurement (in backup)
 - Event analysis (in backup)
- Data Collection
 - Science products
 - Ancillary products
 - Integration control
- Miscellaneous
 - DACs and ADCs (in backup)
 - Autonomous operation



EPI-Lo Application S/W – MCP HV



- EPI-Lo software controls MCPs' High Voltage (HV) common supply (bulk) and four individual HVs (opto-isolator controlled)
- EPI-Lo software slowly ramps voltages to commanded levels
- At least two commands are required to turn on HV
- There are also commands to control current monitors implemented in hardware
- Commands:
 - EPILO_HV_COM_LEVEL: Set Common High Voltage Supply Level
 - EPILO_HV_COM_LIMIT: Set Common High Voltage Supply Limit
 - EPILO_HV_MCP_LEVEL: Set MCP High Voltage Supply Level
 - EPILO_HV_MCP_LIMIT: Set MCP High Voltage Supply Limit
 - EPILO_HV_MCP_STEP: Step MCP High Voltage Supply Level
 - EPILO_HV_CUR_ENB: Enable/Disable MCP HV Current Monitoring
 - EPILO_HV_CUR_LIMIT: Set MCP HV Current Monitoring Limit



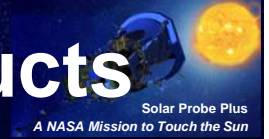
EPI-Lo Application S/W – SSD Bias



- EPI-Lo software controls SSDs' Bias Voltage (BV) supply
- EPI-Lo software slowly ramps the supply voltage to commanded level
- At least two commands are required to turn on bias supply
- Commands:
 - EPILO_BV_LEVEL: Set SSD Bias Supply Level
 - EPILO_BV_LIMIT: Set SSD Bias Supply Limit
 - EPILO_BV_STEP: Step SSD Bias Supply Level



EPI-Lo Application S/W – Science Products



- Ions are collected with different energy resolution, angular (position) resolution, and cadence
 - Ions - Fast: 8 angles, 8 energies
 - Ions - Slow: 80 angles, 69 energies (29 H, 12 He, 14 Heavy group 1, and 14 Heavy group 2)
- Electrons are collected with different resolution and cadence
 - Electrons - Fast: 8 angles, 6 energies
 - Electrons - Slow - Regular: 8 angles, 32 energies
 - Electrons - Slow - Hi-Angle: 80 angles, 16 energies



EPI-Lo Application S/W – Ancillary Products



- Basic rates (i.e. singles) are collected
 - Hardware counters, e.g. SSD, Start1, Start2, Stop, Events queued, etc.,
 - 24-bit hardware counters accumulated in 32-bit software counters
 - Collected per-quadrant or octant
 - Software counters, e.g. Events processed
- Single events
 - Software reads events from hardware FIFO
 - A commandable number collected in raw event product



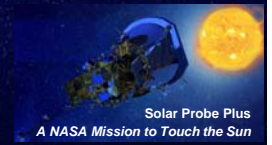
EPI-Lo Application S/W - Integration Control



- EPI-Lo software defines data products as fast or slow
- The time to integrate fast and slow data is commandable
- Each data product can be individually enabled or disabled
- Commands
 - EPILO_DAT_COLLECT - Set Data Collection Pattern
 - EPILO_DAT_ENB - Enable/Disable Data Products
 - EPILO_DAT_RAW - Control Amount of Raw Event Data
 - EPILO_DAT_TIME - Control Data Integration Time



EPI-Lo Application S/W – Autonomous Operations



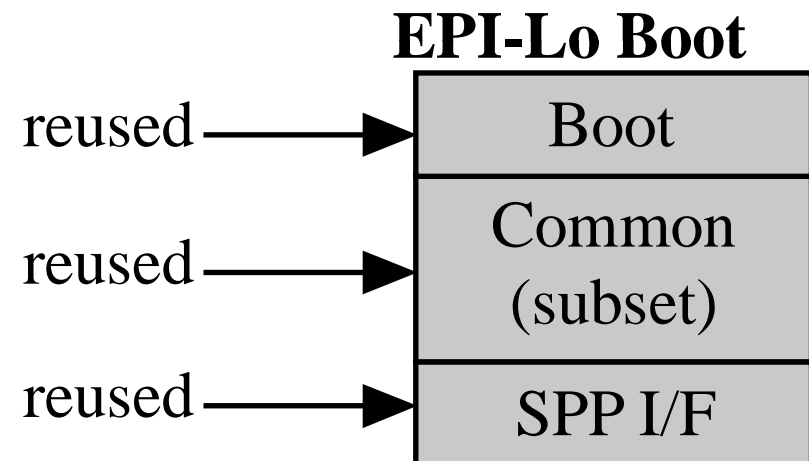
- Common-code macro capability support definition, execution, and save/restore of command sequences
 - Macros can pause until a given MET or delay a given number of seconds
 - Macros can contain do-loops, calls and “forks” to other macros
- Time and status from spacecraft includes
 - Startup Mode: selects manual vs. autonomous operation
 - Solar Distance: current distance to the Sun and in vs. outbound
- If autonomous operation is selected, EPI-Lo software will:
 - Automatically load macros from MRAM
 - Run startup macro; this ramps up HVs, etc.
 - Monitor solar distance against a set of commandable thresholds
 - Different threshold crossings trigger different macros to run
 - Macros configure science collection, e.g. integration times, etc.
- If autonomous operation is not selected, EPI-Lo software will wait for commands



EPI-Lo Boot S/W

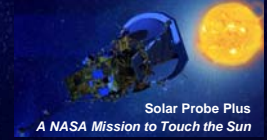


- Commands: memory load, copy, execute, dump, check, plus boot from MRAM
- Telemetry: status/housekeeping, memory dump, etc.
- High (>95%) code reuse:
 - SPP I/F from application
 - Common software (subset)
 - Boot software (boot programs from MRAM)
- New: wrapper, startup, etc.
 - If autonomous operation is selected, EPI-Lo boot software will automatically try to boot a series of programs from MRAM
 - If autonomous operation is not selected, EPI-Lo boot software will wait for commands





Software Development Environment



- Flight software and development tools are on Space Department's Unix system
 - /project/spp-instr
 - Backed up nightly
 - Tools are C-based and have been used on Sun OS, Linux, & Mac OS X
 - Only single developer can modify software
- Software versions
 - Numbered with integers, i.e. 1, 2, 3, etc.
 - Each version is in a subdirectory; name of subdirectory includes version number
 - Version number reported in telemetry
 - Each delivered version includes snapshot of libraries
 - Archived in Product Lifecycle Management (PLM) system
- Testing
 - Unit and integration testing done on Xilinx prototype board (no sensor); concocted events are in RAM
 - Further testing on EM with pulsers simulating events
 - Acceptance testing is done as part of Instrument Test Procedure



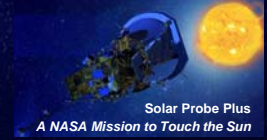
Summary



- EPI-Lo application software
 - EPI-Lo-specific software - new code
 - Common software - reused code, ~50% of application
 - SPP spacecraft interface software - new code, but derived from RBSP interface code
- EPI-Lo boot software, >95% reused code



Backup





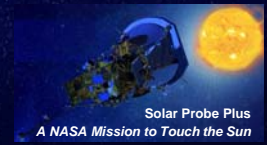
Common S/W - Packet Telemetry



- Provides standard API to application
 - Delivers variable-length packets
 - Numeric ID identifies packet type/contents
 - Time tag of data collection time
- API also used within common code to deliver standard products:
 - Memory dump, memory checksum, etc.
 - Command echo
 - Alarm
 - Status/housekeeping



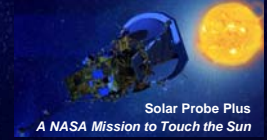
Common S/W - Commands



- Standard API for “clients” to register commands:
 - Numeric “opcode” and expected number of arguments
 - Pointer to code that implements command
- API also used within common code to register standard commands:
 - EPILO_CMD_NULL: do nothing; tests uplink and downlink paths
 - EPILO_STAT_CLR: reset counters
- Command process executes commands
 - Echoes opcode, up to ten argument bytes, and a result code
 - Also executes stored command sequences, macros ...



Common S/W - Macros



- Macros are stored sequences of commands
- 256 different macros can be defined
- 64 Kbyte of RAM is available for macro storage
- Macros can nest 16 deep; up to 64 macros can execute concurrently
- Real-time uplink commands take precedence over macro commands
- Commands from a macro are echoed; the echo includes a flag indicating that the command is from a macro



Common S/W - Macros (cont.)



- Macros are “learned” by the instrument
 - EPILO_MAC_DEF starts a macro definition; any command uplinked with its “macro” arg set will be appended to the macro
 - EPILO_MAC_ENDDEF ends the definition
 - While a macro is being compiled, any real-time command, i.e., one without a set “macro” arg will be executed.
 - There is no need for macro compiler or macro memory management by ground software
- Other macro commands:
 - EPILO_MAC_DELAY and EPILO_MAC_PAUSE delay by a give number of seconds or until a given time, respectively
 - EPILO_MAC_NEST and EPILO_MAC_RUN nest a macro and starts a concurrently executing macro, respectively
 - EPILO_MAC_LOOP_BEGIN and EPILO_MAC_LOOP_END delimit a definite loop
 - EPILO_MAC_HALT kills a running macro



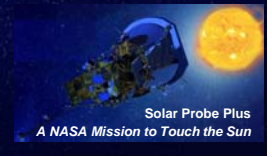
Common S/W - Memory Management



- Commands:
 - EPILO_MEM_LOAD: loads RAM
 - EPILO_MEM_COPY: copy memory; source and/or destination can be RAM or non-volatile MRAM
 - EPILO_MEM_READ: produce memory dump packets
 - EPILO_MEM_READ_ABRT: stop memory dump
 - EPILO_MEM_CHECK: produce a checksum packet summarizing given region
 - EPILO_MEM_RUN: jump to program at given address
- Note: these commands are the core of the boot software



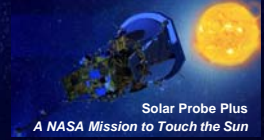
Common S/W - Monitoring and Alarms



- Alarm packet is generated in response to a software problem or to a monitored value going out of limits
- Alarms from monitors can be transient or persistent
 - If a monitored value is out-of-limits just once, a transient alarm is reported
 - If a monitored value is consecutively out-of-limits twice, a persistent alarm is reported and the software may take corrective action
 - If a monitored value is consecutively out-of-limits more than twice, either corrective action is taken again, the shutdown macro is run, or nothing is done, depending on the thing being monitored
- Commands:
 - EPILO_MON_CNTRL enables or disables corrective action



Common S/W - Monitoring Algorithm



- High response (low response is similar):

high once:

 issue transient high alarm

high twice:

 issue persistent high alarm

 if enabled (via EPILO_MON_CNTRL command)

 execute high response macro for this alarm

high more than twice:

 case of monitor class

 current/voltage:

 if enabled (via EPILO_MON_CNTRL command)

 run shutdown macro

 temperature:

 nop

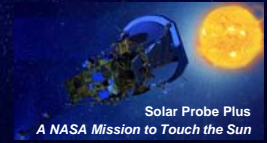
 count rate:

 if enabled (via EPILO_MON_CNTRL command)

 re-execute high response macro



Common S/W - Status Reporting



- Status packet
 - Command and macro execution counters
 - Analogs: voltages, currents, and temperatures
 - Commanded instrument state
- Commands:
 - EPILO_STAT_INT: how often should status be reported



EPI-Lo Application S/W – Time-Of-Flight Subsystem



- Measures particle time-of-flight (TOF), angle (i.e. position) on the sensor, and pulse height
- EPI-Lo software controls discriminator thresholds
- Commands:
 - EPILO_TOF_CFD: Set TOF Constant Fraction Discriminator Threshold
 - EPILO_TOF_THRESH: Set TOF Pulse Height Discriminator Threshold



EPI-Lo Application S/W – Energy Subsystem



- Measures particle energy
- EPI-Lo software time-multiplexes hardware between electron and ion detectors
- EPI-Lo software controls thresholds, one set per detector
- Commands:
 - EPILO_EGY_THRESH: Set Energy Discriminator Threshold
 - EPILO_EGY_ANTI: Set Energy Anti-Coincidence Threshold



EPI-Lo Application S/W – Event Subsystem



- Analyzes data from TOF and energy subsystems to identify an “event”
- EPI-Lo software configures hardware to define a valid event
- Commands:
 - EPILO_EVT_MULTI: Enable/Disable Multiple Hit Reject
 - EPILO_EVT_WINDOW: Set Event Coincidence Window



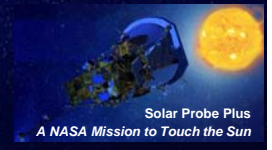
EPI-Lo Application S/W – DACs and ADCs



- 1 Hz activity schedule
- Control QuadDACs over I2C bus
 - Set power supply levels, e.g. MCP HV
 - Set thresholds, e.g. MCP HV current
- Read ADCs over SPI bus
 - Read analogs (read three times and use median value)
 - Save in housekeeping
 - Monitor and report/respond to out-of-limit conditions



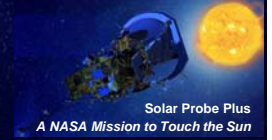
EPI-Lo Application S/W – Test Support



- EPI-Lo software provides commands to control an on-board pulser
- EPI-Lo software provides commands to select internal test points to bring out on the test port
- Commands:
 - EPILO_TST_PUL_CFG: Configure Pulser
 - EPILO_TST_PUL_ENB: Enable/Disable Pulser
 - EPILO_TST_POINT: Select Test Point Signals



Reference Documents



- 7434-9066, “Solar Probe Plus (SPP) General Instrument (GI) ICD”
- 16105-ISIS-IRD-01, “PRELIMINARY INSTRUMENT REQUIREMENTS DOCUMENT SOLAR PROBE PLUS PROJECT ISIS INSTRUMENT”
- 7464-9005, “EPI-Lo Flight Software Specification”
- 7464-9003, “EPI-Lo Software Development Plan”