STEREO
IMPACT

IDPU Software Requirements

IDPUSoftwareRequirements_A.doc
Version A – 2001-Jun-07

David Curtis, UCB IMPACT Project Manager

Document Revision Record

Rev.	Date	Description of Change	Approved By
A	2001-June-7	Preliminary Draft	-

Distribution List

Dave Curtis, UCB
Davin Larson, UCB
Dorothy Gordon, Elf
Tycho von Rosenvinge, GSFC
Mario Acuna, GSFC
Lynn Kistler, UNH

Table of Contents

1. Introduction

The IMPACT IDPU provides the interface between the STEREO Spacecraft C&DH system and the IMPACT and PLASTIC Instruments. The IDPU software runs on a microprocessor in the IDPU, on the Data Controller Board (DCB). All information transfer between the IMPACT/PLASTIC instruments and the Spacecraft/Ground, including telemetry, commands, and status flow through the IDPU. The IDPU performs instrument control functions, telemetry compression and formatting, and system monitoring/safing functions.

This document includes some implementation information in addition to the software requirements. It should contain sufficient information to allow software design and coding to proceed without further documentation steps.

1.1. *Document Conventions*

In this document, TBD (To Be Determined) means that no data currently exists. A value followed by TBR (To Be Resolved) means that this value is preliminary. In either case, the acronym is typically followed by a code such as UCB indicating who is responsible for providing the data, and a unique reference number.

1.2. *Applicable Documents*

The following documents include drawings and STEREO Project policies, and are part of the Interface Requirements. In the event of a conflict between this ICD and the following documents, this ICD takes precedence. All ICD documents and drawings can be found on the Berkeley STEREO/IMPACT FTP site:

http://sprg.ssl.berkeley.edu/impact/dwc/

1. DRAFT_ICD_IMPACT_a (IMPACT/Spacecraft ICD, on the APL web page)
2. ICD/Impact Serial Interface
3. Specifications/IDPUSpec_C
4. Phase A Report/PAIP (Performance Assurance Implementation Plan)

2. Host System

This section briefly describes the environment in which the software will function. More details on what the software does with these capabilities is described in section 3 and 4.

The IDPU software runs on a UT80CRH196KD microprocessor on the Data Controller Board as described in Reference 3. A more detailed description including such things as I/O addresses and bit meanings will be provided when the DCB design matures (TBD-Elf-001)

Users manuals for the processor and 1553 interface chip can be found on-line at the UTMC web site.

2.1. *Code Memory*

Software shall reside in two memories: the boot PROM and the EEPROM. The boot PROM is not modifiable once installed in the DCB, while the EEPROM is modifiable both pre- and post-launch by ground command. The processor starts out on reset executing the code in the boot PROM. The code may either be run out of PROM/EPROM, or be copied into RAM and then run out of RAM (TBD-UCB-002).

2.2. *Memory Map*

The microprocessor addressing space is 64Kbytes. The memory includes 8Kbytes of PROM, 128Kbytes of EEPROM (TBR-UCB-003), and 2Mbytes of RAM. The DCB includes a paging system to map these physical memories into the processor addressing space as described in Reference 3. Processor memory is divided into 4 16Kbyte pages each of which can be mapped to any of the physical memory pages. The software shall control this paging system.

It is expected that one page shall be used for code, while the rest shall be used to access memory buffers such as input data buffers, averaging buffers, burst memory buffers, etc. The microprocessor includes 1K of internal memory that should be sufficient for global variables, stack, etc.

2.3. *Direct Memory Access*

The DCB includes a DMA system to transfer data directly from the interfaces into memory, as described in Reference 3. The DMA services the 1553 interface chip as well as the four serial instrument interfaces. The software must set up this system, selecting buffer addresses and enabling transfers. The software must then keep up with the flow of data to prevent the system from losing or over-writing data.

2.4. *Interrupt System*

The DCB provides two interrupts: One from the 1553 system, and one periodic interrupt. The 1553 interrupt is actually a combination of a number of maskable sources in the 1553 interface chip. The periodic interrupt is intended to provide the basis of a regular polling system for time-critical functions (including the serial instrument interface). In addition the processor has a number of internal interrupt sources, most of which will not be used.

2.5. *I/O System*

The I/O for this processor is memory-mapped. In addition to a number of internal I/O registers that control the internal processor functions, the DCB will implement I/O functions to control external features such as the 1553 and serial instrument interfaces.

2.6. *1553 Interface*

The DCB includes a 1553 interface circuit as described in reference 3. This circuit interfaces the processor system to the spacecraft C&DH 1553 bus. The circuit takes care of most of the low level 1553 issues. Software must initialize the 1553 circuit, allocate it memory space for its buffers and control tables in the processor RAM (which it accesses by DMA), and then respond to the messages as described in reference 1 and the circuit data sheet.

The 1553 interface also supports data exchanges with SWAVES using RT to RT transfers. These shall be exchanged 5 times a second, on intervals of 200ms +/-10ms, without handshake.

2.7. *Serial Instrument Interfaces*

The DCB includes 4 serial instrument interfaces as described in reference 3. These interfaces are used for the IDPU to interact with instruments (one interface each for SWEA/STE, SEP, MAG, and PLASTIC). The processor can send commands and receive telemetry over these interfaces as detailed in reference 2, and further detailed in the IDPU/Instrument ICDs (reference TBD-UCB-004). Telemetry blocks are transferred by DMA into a circular buffer in processor RAM. Note that this interface has no handshake; the software must deal with messages from the instruments as fast as they are sent, before the circular DMA system over-writes the data. The system will generate an error flag if an over-write occurs.

2.8. *Timing*

The C&DH system provides UTC time, but with some jitter. The 1553 interface circuit provides a scheme that will allow extracting this time code without adding additional jitter by time-tagging message receipt. The software shall maintain a UTC time code based on this interface and the 16-bit sub-second counter in the 1553 interface circuit.

A separate sample clock based on the DCB oscillator (also the clock source for the processor, 1553, and instrument serial interfaces) shall be used for jitter-free data sampling. Instruments receive this clock as described in Reference 3, so all sampling will be synchronous. Circuitry in the DCB maintains a sub-second and seconds counter that the software can read via I/O.

2.9. *Watchdog Timer*

The processor includes a watchdog timer system that will be used to detect software crashes and reset the processor.

2.10. *Diagnostics*

The processor includes a UART that may be used for early software diagnostics. It will not be used during system integration or in flight. This will be brought out on an internal test connector, which also carries processor bus signals and FPGA diagnostic signals to aid in software and hardware debugging.

2.11. *Processor Idle Mode*

The processor has a low power idle mode. This should be used whenever the system has no work to do in the foreground module. The next interrupt will bring the processor back out of idle mode; when the interrupt handler completes, the foreground should poll all modules for foreground tasks and then return to idle. (Note that this assumes a operational system structure consisting of a polled foreground system. This is not required; if some other system is used, a similar scheme for going to idle mode when no task is pending should be used to save power). This allows the use of a faster processor clock without incurring all of the power cost.

3. Service Software Requirements

3.1. *Operating System*

A simple operating system needs to be developed to divide processing resources amongst modules. Memory resources shall be static for simplicity; only processor cycles are shared. Modules shall be self-contained with minimal interaction. They shall receive processing time either as part of an interrupt handler or in the foreground as part of a polling loop. Tasks requiring more time than is consistent with the interrupt handler must be passed off to the foreground loop. It is believed that no prioritization of tasks or other more complex scheduling than this shall be required (TBR-UCB-005).

3.2. *Boot PROM*

The Boot PROM software is run on processor reset. It shall contain a subset of the full code sufficient for its purposes as described here.

The software first initializes the system, including the 1553 command/telemetry system (initialization of the instrument interfaces is optional). The boot software need generate only housekeeping telemetry and shall respond to a subset of the instrument commands (memory load).

The software then decides if the code in the EEPROM is valid (based on a checksum or similar scheme), and if so, to transfer control to it. A short time delay (few seconds) shall be added before transferring to EEPROM to allow for the possibility of an override by command from the spacecraft. This allows the shift to EEPROM to be aborted if desired.

The Boot PROM shall be able to load memory code (RAM or EEPROM) from ground command, so that software can be re-loaded in case the EEPROM is corrupted.

EEPROM code can be either transferred to RAM or executed out of EEPROM (TBD-UCB-006)

After the PROM has transferred control to the EEPROM, it should be powered off to save power.

The Boot PROM may include more of the functionality nominally found in the EEPROM if such code is available at the time the PROM is burned.

3.3. *Spacecraft C&DH Interface*

The software must initialize the 1553 interface circuit as called out in the part datasheets, and consistent with the requirements called out in Reference 1. Once initialized, the software must respond to messages and error conditions generated by the interface. The polling and/or interrupt handler used to respond to errors and messages need to be fast enough to avoid losing data or telemetry transmission opportunities in the worst case conditions (for example, 10ms between telemetry blocks).

3.4. *Command Routing*

The C&DH interface shall provide commands in CCSDS packets as described in Reference 1. The command router shall use the ApID to determine which module is to receive the command packet. Each module may be allocated one or more ApIDs for commands.

Most commands are expected to be short in length (few bytes). Give the limitation of uplink bitrate and the large overhead in packet and other headers, a system to pack multiple commands into a packet may be considered (TBR-UCB-007).

3.5. *Telemetry Packet Queue*

Modules shall provide formatted CCSDS telemetry packets to the packet queue. The queue shall provide the packets to the C&DH interface. A bit-counting scheme shall be used to maintain the packet rate to the C&DH system at the average allocation (this allocation value shall be programmable by ground command). Each module shall be responsible for generating packets at its allocated average rate over the long term, and shall not exceed its average packet rate by more than 4 (TBR-UCB-008) packets per module at any time. A queue over-run will generate an error message and result in loss of data. This will limit the required size of the packet queue and eliminate the need the system to know the allocations of each module. Note that some long accumulations will generate occasional large blocks of packets. If the queue size is insufficient to meter out these blocks, the modules need to manage this buffering themselves. Any unused telemetry space (based on the bit counting scheme) shall be used to play back burst telemetry blocks.

3.6. *Timing*

Telemetry packets shall be time-stamped with the UTC time code. Normally this time code will correspond to the collection time of the first sample in the packet, but any similar scheme which allows the ground software to unambiguously correlate samples to UTC is acceptable.

Data accumulation is actually done using the sample clock. The ground software can determine the relative timing of data in a jitter-free manner by determining sample number based on the packet sequence counter. The housekeeping packet shall also contain information to allow a correlation between UTC and sample time.

To the extent possible, sampling and accumulation intervals of different instruments should be synchronized. The software should synchronize averages to appropriate sample clock intervals. A binary minute (64 seconds) may be used as a time standard for longer averages (TBR-UCB-009).

3.7. *Burst System*

The burst system is used to capture high time resolution data during "events", and play it back slowly as part of the IMPACT telemetry system. Instrument modules provide a continuous stream of high time resolution data to the Burst system, which records it in a circular RAM buffer. A criterion function based on IMPACT, PLASTIC and SWAVES data indicates when an event occurs and how good that event is. The best event over the interval required to play back the previous burst is then sent as the next burst. This involves three buffers; one for the transmitting burst, one for the current best burst, and one for the current data while looking for a better burst. Most of the 2Mbyte RAM shall be used for these Burst buffers. Burst data shall include data both before and after the peak sample of the burst criteria function. Burst data to be saved shall be passed to the burst system by the instrument modules in the form of CCSDS telemetry packets.

The criteria functions computation is considered part of the Instrument software. The burst criterion is then passed to the Burst system, which uses it to decide what data is kept for transmission.

3.8. *Serial Instrument Interfaces*

The software must setup the serial instrument interfaces by allocating a buffer for received data blocks (DMA). Sample Time commands are sent automatically, but the software must forward UTC commands to SEP for packet headers in a manner that minimizes additional timing jitter. The instrument modules should respond to new data blocks and send instrument mode commands as needed. The software must also meet any special command timing requirements for the instruments. For example, SEP cannot receive commands closer together than 10ms (TBR-SEP-010).

3.9. *Housekeeping*

The software must generate a housekeeping packet from information provided by the Instrument modules, plus some executive status information such as command verification, reset counter, error reporting, etc. A single housekeeping packet combining all the IMPACT and PLASTIC housekeeping is preferable to maximize the sample rate with minimum bitrate allocation (32bps allocated for IMPACT, TBD for PLASTIC).

A high rate housekeeping mode for diagnostics is desirable if it does not complicate the instrument interfaces too much, perhaps 8x the nominal rate. This mode would be enabled by ground command.

3.10. *Beacon Data*

The IDPU shall collect beacon mode data from the instruments and format it into a common packet every 64 seconds (TBR-UCB-011). Each Instrument Software module shall format its share of this block and pass it on to a common model to be formatted into a single packet and transferred on to the spacecraft with ApID TBD-UCB-012.

3.11. *Error and Command Verification Log*

A log of error messages and commands received shall be collected in a diagnostic packet and transmitted when full or after a minute of no new entries. Entries shall be timetagged with the UTC time provided by the spacecraft, and shall include all relevant information. Command verification messages for memory loads need not reflect the full command packet; a subset including identification information from the packet header should be sufficient.

3.12. *Safing*

Safing refers to any autonomous function the system requires to put the instrument in a safe mode in the event of an off-nominal condition. The conditions to be monitored include the spacecraft status (in particular shut-down warnings and thruster firing warnings), as well as instrument housekeeping. The list of items to monitor and default actions to be taken is included in Table 3.12-1. The system shall be table driven to allow flexibility.

Table 3.12-1 Safing Table (TBR-ALL-099)

Item	Monitor	Action
1	Shut-down warning	Power off HVPS
2	Thruster Firings	Shut down TBD HVPS, shut STE door
3	Over temperature	Shut down subsystem
4	HV Arc	Shut down HVPS
5	…	

3.13. *Reliability*

The software must be robust in handling errors or failures. Whenever possible (consistent with instrument safety) the system should return to an operational state following an error and not wait for intervention from the ground.

The hardware watchdog timer shall be reset in the periodic timer interrupt handler. That module in turn shall keep track of the status of all other modules using a software watchdog scheme.

On reset the system shall start up automatically and begin operating in a default mode. Instruments that require high voltage shall have to wait for ground command to bring the high voltage back on.

The system shall be as modular as possible, with as little interaction between modules as possible. This is particularly true for PLASTIC; the PLASTIC team should be able to operate the PLASTIC instrument independent of the state of IMPACT and vice versa.

Reference 4 includes some performance assurance requirements related to software that shall be met by the software development.

4. Instrument Software

Instrument software shall be divided into modules with minimal interaction to simplify development, test, and maintenance.

4.1. *MAG*

4.1.1. Sample Data Input

MAG interfaces to the DCB via a Serial Instrument Interface (reference 2). MAG samples are provided in blocks of 4 16-bit words with no header (a deviation from reference 2, but reasonable since only one kind of data is sent). This sample includes the 3 MAG axis samples (x,y,z) and one IDPU analog housekeeping data sample, in that order. MAG data is sampled at 32Hz synchronous with the sample clock. MAG data will be transferred over the interface shortly after it is sampled.

The analog housekeeping shall sample one of 8 analog values in rotation, synchronized to the sample clock. The voltages sampled include the LVPS outputs (+/-12V, +5, +2.5), plus TBD-MAG-013 (MAG temp, heater power?). These 8 values shall be included in the IMPACT housekeeping packet, together with a status information such as calibration mode, range, MAG telemetry sample rate, etc.

4.1.2. Instrument Commands

A single command word shall be recognized by the MAG hardware (in addition to the automatically sent sample clock synchronization command). This command shall include a range bit and a cal mode bit. The command value shall be latched into the MAG system on the next 1Hz sample time tic; the MAG sample for that tic will use the new settings. The software should synchronize sending this command so that the transition takes place at the start of a new telemetry packet.

The Cal mode bit shall be a simple pass-trough from ground command.

The MAG software module shall also accept ground commands to set parameters such as ranging thresholds and packet sample rates.

4.1.3. Ranging

The ranging will have two modes selected by ground command: fixed on or off, and automatic. In automatic mode, the software will decide once a telemetry packet what range the MAG should be in based on the samples collected, using the following rules:

1. If in high range, and all samples from the previous packet are less than threshold value DOWNRANGE, set to low range.
2. If in low range and any sample from the previous packet is greater than threshold value UPRANGE, set to high range.

4.1.4. Telemetry Packets

MAG telemetry packets shall consist of 43 MAG samples (43 * 3 axis * 2 bytes = 258 bytes) plus one 2-byte status field, and the 12-byte CCSDS packet header, for a total of 272 bytes. The status field shall include the current CAL and Range bit settings, a sample rate code, plus TBD-UCB-014.

The CCSDS packet header time stamp will contain the UTC time value for the first sample (with whatever jitter there is in the UTC time system). The packet sequence counter shall be used for jitter-free relative sample clock timing.

The normal telemetry MAG values shall be averaged from 32 Samples/second down to 4 Samples/Second in the default mode. The averaging interval shall be programmable so that higher sample rates can be generated during diagnostics or other times when higher bitrate is available. Burst MAG packets shall use the full 32 samples/second.

A compressed MAG telemetry scheme may be added later to provide higher sampling rate within the same telemetry bandwidth (TBR-UCB-015).

4.1.5. MAG Beacon Data

MAG beacon data consists of one 3-axis sample (6 bytes) per beacon accumulation interval. This value shall be averaged over the sampling interval. In addition one bit of ranging status information shall be included. If the range changes during the averaging interval, only the data up to the time of the range change shall be averaged into the data (this is never expected to happen on-orbit).

4.2. *SEP*

SEP instrument control and data formatting is performed in the SEP processor(s). The IDPU is primarily a bent pipe for telemetry packets and command packets between the spacecraft and SEP.

4.2.1. SEP Commands

Commands with ApIDs in the range allocated to SEP (TBR-UCB-016) shall be passed on to SEP via the SEP serial interment interface as described in Reference 2. The IDPU shall buffer the command packets and meter out the words at no more than one 24-bit interface command word each 10ms (TBR-SEP-010) to avoid swamping the SEP processor. Note that the SEP processor must deal with the automatic sample timing commands sent once a second separately, since the IDPU processor does not control it.

SEP also needs the UTC information provided by the spacecraft in order to time-stamp its CCSDS packets with collect time. Time information shall be forwarded over the SEP serial instrument interface as described in Reference 2. This sequence may or may not be subject to the command rate limitation (TBR-SEP-017); if so, the UTC timing will have more jitter.

A reset command to SEP shall be sent via the instrument serial interface by the IDPU when requested by ground command.

Other information passed to SEP (i.e. spacecraft status) is TBD-SEP-018.

4.2.2. SEP Telemetry Packets

SEP shall provide CCSDS formatted packets over the instrument serial interface as described in reference 2. SEP telemetry packets shall have message type code 0. These packets are provided without handshaking at a rate determined by SEP. SEP is responsible for ensuring that the rate at which they send packets does not exceed their bitrate allocation and the buffering scheme described in section 3.5 above. The SEP bitrate can be programmable so that higher bitrates can be generated for diagnostic or other reasons when more telemetry is available (either because the spacecraft has provided more bitrate to IMPACT, or because other instruments in IMPACT have decreased requirements. This will need to be coordinated on the ground. If too much SEP telemetry is sent and the telemetry packet queue is filled, telemetry packets will be lost.

SEP telemetry packets shall have ApIDs in the range TBD-UCB-019. ApIDs should be unique so ground processing software can route the data and determine how to process the data with no information not in the telemetry stream.

SEP currently makes no Burst data (TBR-SEP-020).

Any special SEP diagnostic packets (memory dumps, error messages, etc) can be sent using this same packet scheme. The SEP daily bitrate allocation can be violated by a few percent for this purpose without serious consequences provided the telemetry queue limitations described in 3.5 are not violated.

4.2.3. SEP Housekeeping

SEP shall collect housekeeping information from the SEP instruments including voltages, temperatures, digital status and modes, command verification, and error messages, into a block 64 bytes long (TBR-UCB/SEP-021). This shall be sent to the IDPU over the serial digital interface with message type code 01. This block shall be sent once a second (TBR-UCB-022). The timing of the sampling of data in this block should be deterministic so that it can be reconstructed on the ground to the precision required by SEP. The IDPU shall combine the most recent version of this housekeeping block into the IDPU housekeeping packet when it is generated. Note that the IDPU housekeeping packet rate is programmable, but the SEP housekeeping block rate is fixed, so that the SEP housekeeping is sub-sampled. SEP should not assume all housekeeping blocks generated get to the ground.

4.2.4. SEP Beacon Data

SEP shall provide formatted beacon mode data once a beacon accumulation interval to the IDPU over the SEP serial instrument interface. The accumulation interval shall be synchronized to the sampling clock modulo the accumulation interval. The beacon mode data message shall have message type code 02 and shall be 46 bytes long (TBR-UCB-023). It shall be provided no later than 1 second after the accumulation interval is complete.

4.3. *SWEA*

The SWEA instrument interfaces via the SWEA/STE Serial Instrument Interface described in reference 2. SWEA contains no processor and so is interfaced with at a low level.

4.3.1. Instrument Control

SWEA instrument control is automated. The processor need only load some look-up tables and control registers, and the instrument then proceeds on its own. The values to be loaded are based on a default set of values in EEPROM or else a modified set of values loaded by ground command. The format for the look-up tables are registers is TBD-UCB-024.

High voltage supply control will involve software interlocks in addition to the hardware enable plug. Turning on the high voltage supplies shall require a minimum of two instrument commands to provide some additional safety (turning on high voltage at the wrong time can damage the instrument).

When controlling the high voltage supplies, software shall limit the ramp-up rate to less than 200V/sec by stepping the supply level up slowly.

4.3.2. Data Collection

SWEA shall generate a block of 16 16-bit counters. In addition a word indicating when in the 2-second sweep cycle the data was accumulated will be included in the block which can be used to identify the sweep settings corresponding to the accumulation. This data shall be sent over the serial instrument interface along with the STE data and the housekeeping data. The telemetry blocks sent over this interface shall conform to the standard in reference 2, and the SWEA counter blocks shall have Message ID=00. This will allow the software to parse through the SWEA/STE message DMA buffer and identify the SWEA counter blocks.

Counter blocks shall be sent automatically every major step of the deflector/analyzer supplies, every 6.4ms (the deflector supplies shall step more often than once an accumulation to integrate over elevation angle).

The first step the processor will perform on this data is to sort it into a distribution buffer covering 50 energies by 6 deflector settings by 16 anode counters. One distribution buffer will be generated every 2 second cycle, and reduced into telemetry products in the next 2 second cycle, so a double-buffering scheme may be used.

4.3.3. Moments

The software shall compute a set of 10 moments of each distribution buffer (every 2 seconds). Moments shall include density (N), three components of velocity (NV), and six components of the pressure tensor (NP). The values shall not be normalized by density onboard; the sum of products shall be sent. Trigonometric and energy terms shall be obtained from tables in EEPROM. Anode and energy dependent efficiency factors shall be applied (these may be applied separately or may be factored into the energy/angle coefficient table, whichever is most efficient). The order of computations can be optimized for computational and/or memory requirements. Fixed-point computations can be used, but care must be taken to maintain enough resolution and dynamic range. The final accumulations shall be compressed into a TBD-UCB-025 16-bit floating-point format for transmission to the ground. The compression scheme shall optimize resolution (maximum number of mantissa bits), based on scientific limitations of the dynamic range (TBD-UCB-026).

The 20-byte moment computations for each distribution shall be accumulated sequentially into a CCSDS packet. 13 moment blocks fit into a packet, so a moment packet shall be generated every 26 seconds. The time stamp on the packet shall correspond to the start of the accumulation of the first set of moments in the packet.

4.3.4. Pitch Angle Distributions

Pitch angle distributions are made by sampling the polar (Deflector) angle distribution along the azimuth (Anode) angle of the magnetic field measurement. The Magnetic Field direction at the start of the accumulation of the distribution buffer shall be used. The telemetered MAG values shall be used so the raw MAG vector used can be found in a MAG packet on the ground. The MAG azimuth angle shall be computed using MAGx, MAGy, and an arctangent lookup table.
The 12 polar samples shall be extracted from the distribution from the azimuthal direction corresponding to the MAG field vector (and its inverse). These shall be sampled over 7 energy range averages. These energy ranges shall be programmable, but default to TBD-UCB-027. Data shall be sampled periodically at a programmable rate (defaults to every 8 seconds). The data shall be compressed to 8 bits per sample using a standard 19 to 8 bit log compression scheme. Along with this data shall be included the computed MAG azimuthal angle. This produces 12*7+1=85 bytes of data. Three consecutive blocks of data shall be included in a Pitch Angle packet. The few extra packet bytes can be used to describe the energy binning scheme used.

4.3.5. Distributions

Distributions shall be extracted from the Distribution Buffer by summing counters into 16 programmable energy accumulation intervals (default intervals TBD-UCB-028) by 80 angle accumulations. The angle accumulations are made by summing the most polar azimuthal samples in pairs (to avoid over-sampling the polar regions). The data shall then be compressed to 8 bits per sample using the same scheme used by the Pitch Angle distributions. This generates 16*80 = 1280 bytes per distribution, which shall be transmitted in 5 telemetry packets. The few extra bytes shall be used to sequence the 5 packets and to describe the energy binning scheme used.

The Distributions shall be sampled (not averaged) at programmable intervals (default = 64 seconds).

4.3.6. Burst Telemetry

Burst telemetry from SWEA shall consist of distributions (see 4.3.5) sampled every 2 seconds.

4.3.7. Beacon Telemetry

Beacon telemetry shall consist of moments data (see 4.3.3) sent once per beacon packet (64 seconds).

4.4. *STE*

The STE instrument interfaces via the SWEA/STE Serial Instrument Interface described in reference 2. STE contains no processor and so is interfaced with at a low level.

4.4.1. Instrument Control

The STE interface includes a look-up table containing the measured energy to energy bin conversion. This table and other control registers such as threshold DACs must be initialized and maintained periodically by the IDPU via the serial instrument interface from default values in IDPU EEPROM.

4.4.2. Data Collection

STE shall send down a block containing 128 16-bit counters every 2 seconds. These counters cover (typically) 8 detectors by 16 energies, though the energy look-up table can change this allocation. This data shall be extracted from the SWEA/STE interface telemetry stream based on its message ID=01.

4.4.3. Telemetry Formatting

The software shall average the 128 counter arrays counters over a programmable accumulation interval (default = 16 seconds), compress them to 8 bits per sample using the standard log compression scheme, and place 2 such accumulations in a STE packet. Spare bytes can be used to describe the LUT and threshold settings.

4.4.4. Burst Telemetry

Burst telemetry is the same as normal telemetry, but sampled every 2 seconds (not averaged).

4.4.5. Beacon Telemetry

Beacon telemetry shall be 8 of the 128 accumulators, averaged over a 64-second accumulation interval and compressed to 8 bits. The 8 values shall each consist of a sum over a TBD-UCB-029 range of the 128 accumulators, nominally covering 4 energies by 2 angles.

4.5. *PLASTIC*

TBD-PLASTIC-100

4.6. *SWAVES Interface*

4.7. *Burst Criterion*