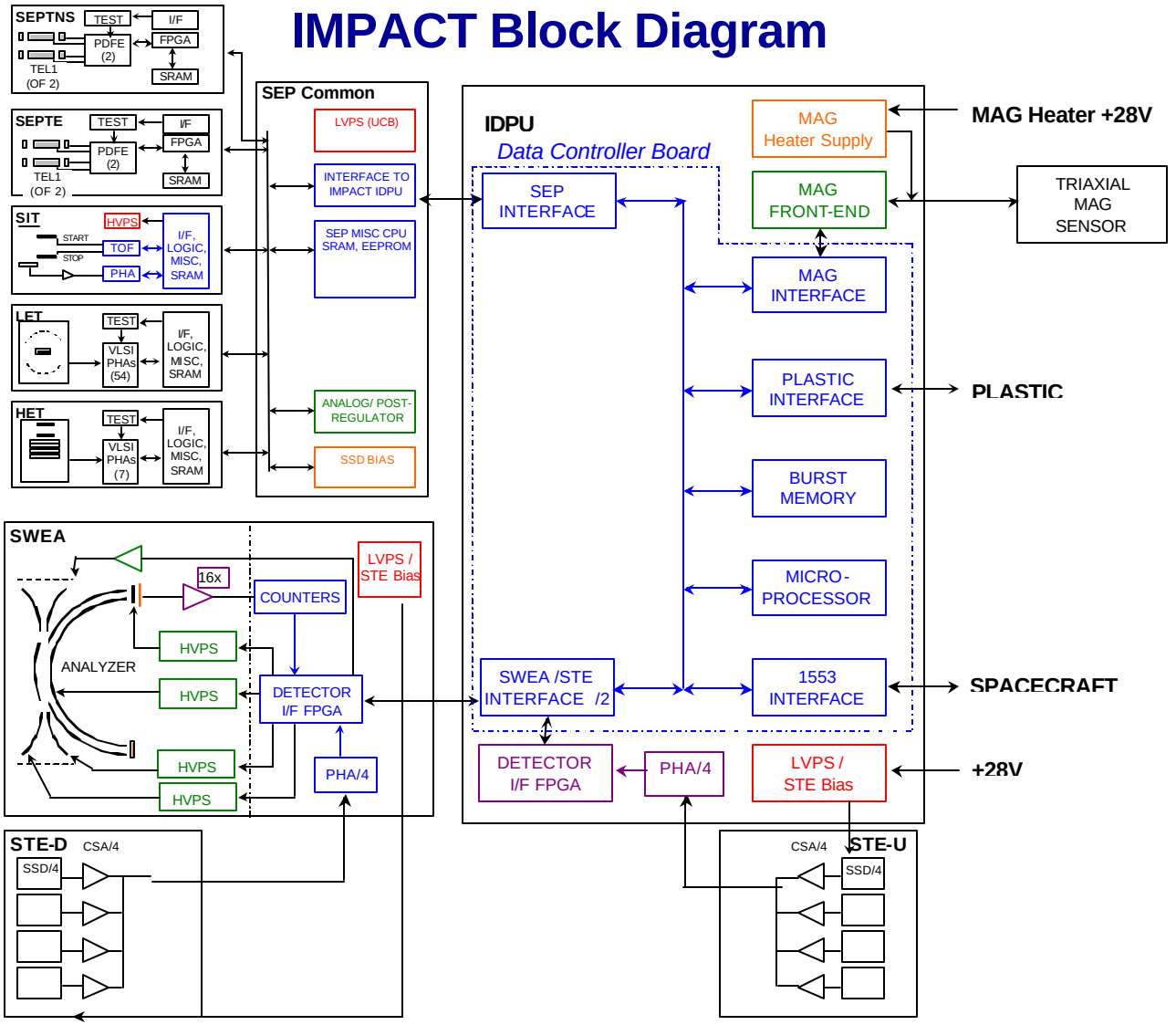
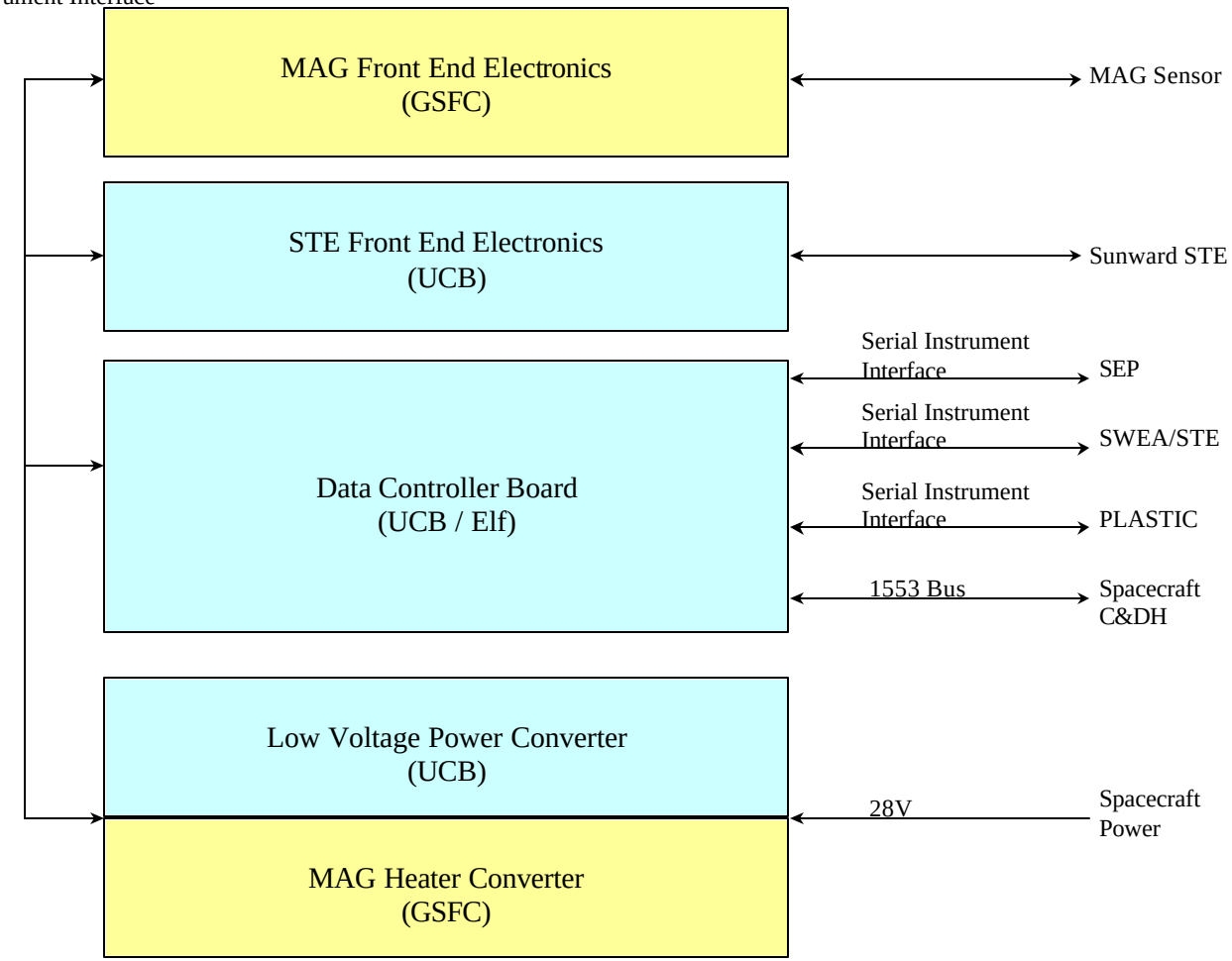
IDPU Overview

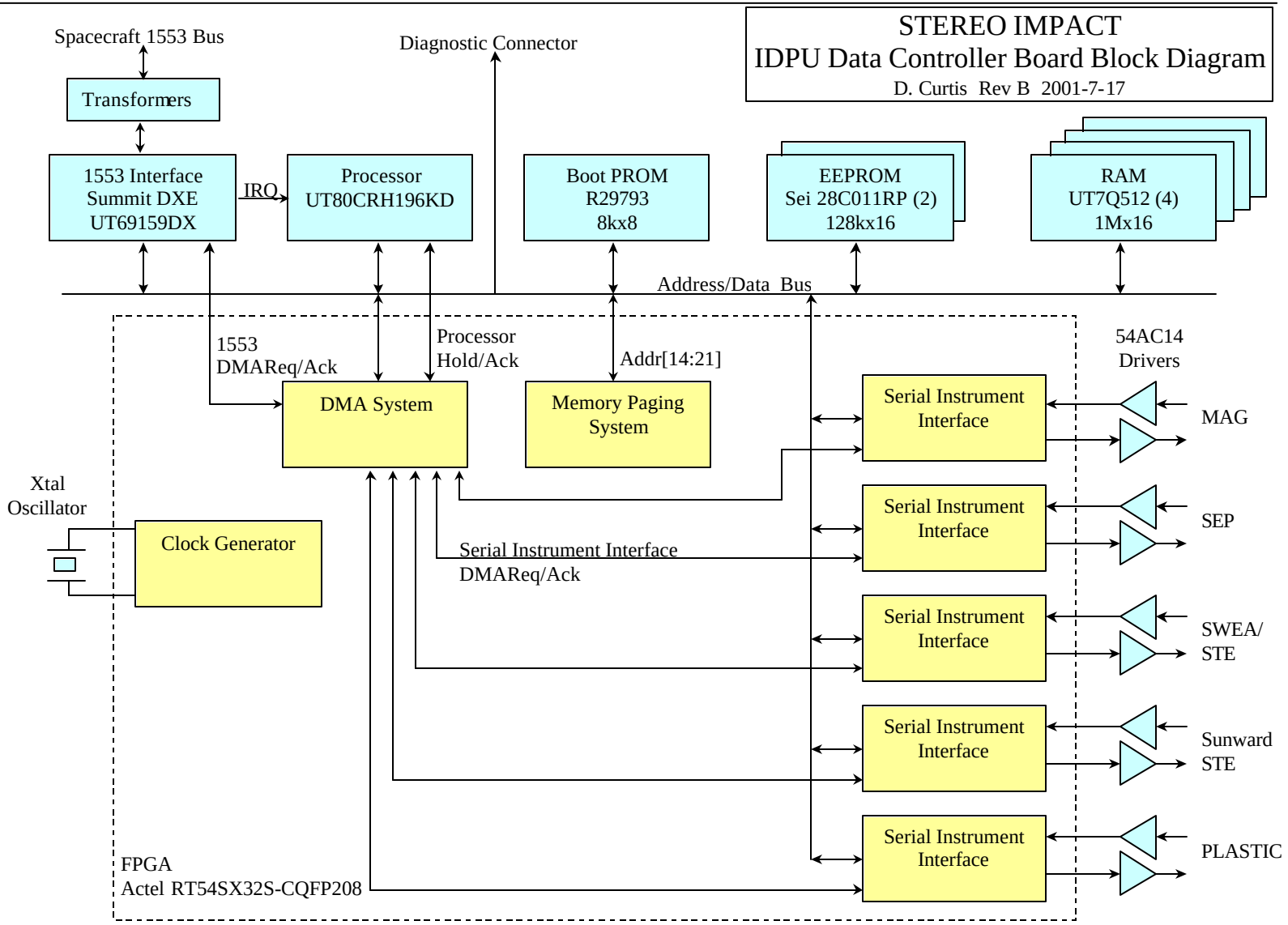
IMPACT Block Diagram



"Stackable Connector" Interconnect:
+/-12V, +/-5, +5V, +2.5V
MAG Serial Instrument Interface
MAG Heater Control
STE Serial Instrument Interface

STEREO IMPACT
IDPU Block Diagram
D. Curtis Rev B 2001-7-17

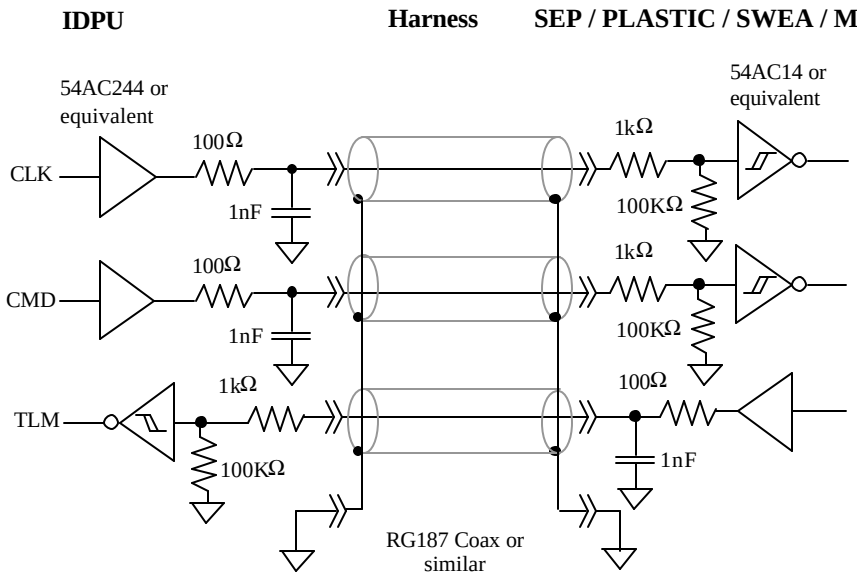


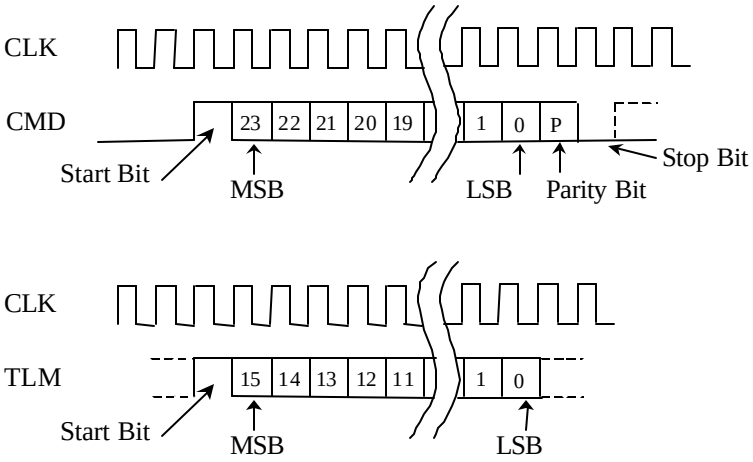

STEREO IMPACT
IDPU Data Controller Board Block Diagram
D. Curtis Rev B 2001-7-17
Spacecraft 1553 Bus
Transformers
1553 Interface
Summit DXE
UT69159DX
IRQ
Processor
UT80CRH196KD
Diagnostic Connector
Boot PROM
R29793
8kx8
EEPROM
Sei 28C011RP (2)
128kx16
RAM
UT7Q512 (4)
1Mx16
Address/Data Bus
1553
DMAReq/Ack
Processor
Hold/Ack
Addr[14:21]
DMA System
Memory Paging
System
54AC14
Drivers
Serial Instrument
Interface
MAG
SEP
SWEA/
STE
Sunward
STE
PLASTIC
Xtal
Oscillator
Clock Generator
Serial Instrument Interface
DMAReq/Ack
FPGA
Actel RT54SX32S-CQFP208

IDPU Hardware Features

- 16-bit processor with high speed multiplier
  - required for moment calculations
- 8Kbyte boot PROM
- 256Kbyte EEPROM for code and tables
- 3Mbytes RAM
  - Mostly for Burst memory, plus data buffering
- Watchdog Timer
- 1553 Interface
  - Message transfer into processor memory buffer automated by 1553 hardware
  - Processor interrupt on error or completed message transfer
- Sample clock interrupt (128Hz)
  - Used for timekeeping, time-critical tasks
- Serial Instrument Interfaces
  - Automatic transfer of instrument data into message FIFOs
- Memory paging
  - maps physical memory into 64K processor addressable memory space in 4 16kbyte pages
  - Uses processor INST signal to allow page 0 (code page) to be 64Kbytes.

Serial Instrument Interface





- **Commands include 8 bit destination code and 16 bits of data**
- **Telemetry messages consist of 16-bit words bounded 17 zeros**
  - **Up to 1025 words per message**
  - **The first word of the message includes a message type code and a message length count**
- **No handshaking is used**

IDPU Software Requirements

IDPU Software Requirements

- First draft of Software Requirements Document June 01
  - Rev B, October 01; Rev C, April 02
- First draft of PLASTIC software requirements from UNH 6/01
  - Rev B, November/01; Rev C, August 02
- Top level requirements:
  - Support Spacecraft 1553 Interface per the Instrument ICD
  - Support Instrument Interfaces per the hardware and software specification
  - Collect, compress, and format telemetry data into CCSDS packets
  - Pass on mode-setting commands from the ground to instruments
  - Limited automation:
    - Set SWEA offset voltage to track spacecraft bias voltage
    - PLASTIC solar wind tracking and entrance system selection
    - MAG ranging
  - Instrument safing (response to HV anomaly, and spacecraft thruster/power down notifications)
  - Support a Burst memory system to collect high time resolution data for short intervals based on a burst trigger criteria
  - Share burst trigger data with SWAVES via RT to RT 1553 transfers

1553 Interface Requirements

- Support the interface protocol as described in the ICD
  - Receive, decode, and forward command packets
  - Send telemetry packets
    - Normal telemetry, Beacon telemetry, Housekeeping telemetry
  - Receive, decode, and synchronize local clock to UTC time code
  - Receive and act on spacecraft status
    - Power down warning; safe instruments
    - Thruster firing warning; safe instruments
    - SSR status – send only state of health when SSR partition almost full
  - Send instrument status
    - Power-down request (on fault detection)
  - Respond to Loopback, TBD mode codes
- Support 1553 Interface chip
  - Setup & maintenance
  - Interrupt on message in/out
  - Error response

MAG Software Requirements

- Average MAG data to 4 samples/second, format into CCSDS packets
- Format 32 sample/second data into CCSDS packets for Burst memory
- Average MAG data to 1 minute and pass to Beacon telemetry formatter
- Perform ranging functions to select one of two gains based on previous measurements (only expected to be used on the ground)

SEP Software Requirements

- Pass on all SEP command packets (by ApID) and Spacecraft Time via Serial Instrument Interface
- Pass on all Telemetry Packets received via Serial Instrument Interface to Telemetry packet queue
- Pass SEP Housekeeping data received via the Serial Instrument Interface to Housekeeping packet formatter
- Pass SEP Beacon data received via the Serial Instrument Interface to Beacon packet formatter

STE Software Requirements

- Initialize and periodically reload STE Energy to Accumulator look-up table via Serial Instrument Interface from IDPU EEPROM
- Initialize and periodically reload STE threshold DAC values via Serial Instrument Interface from IDPU EEPROM
- Log-compress counters and format into CCSDS packets to pass to the Burst system (Spectra and Monitor Rates)
- Time Average and Log Compress counters and format into CCSDS packets for real-time telemetry (Spectra and Monitor Rates)
- Time and Space Average and Log Compress counters and pass to Beacon telemetry formatter (Spectra and Monitor Rates)

SWEA Software Requirements

- Initialize and periodically reload SWEA voltage waveform look-up table via Serial Instrument Interface from IDPU EEPROM
- Initialize and periodically reload SWEA control registers via Serial Instrument Interface from IDPU EEPROM
- Accumulate raw counter measurements into a three-dimensional distribution measurement
- Compute Moments, Pitch Angle Distributions and Full 3D distributions with desired time and space resolution and format into CCSDS packets for the Burst, Real Time, and Beacon telemetry streams
- Adjust SWEA bias voltage based on measured distribution function to offset spacecraft charging effects

PLASTIC Software Requirements

- **Instrument Control Tasks:**
  - **Decoding and routing instrument mode and table load commands**
  - **Controlling the energy at which apertures are switched based on measured count rates**
  - **Controlling "Solar Wind Tracking Mode" when Bursts are triggered to get high time resolution data**
  - **HV safing in response to an Arc**
- **Data Tasks: Collect, compress, and format**
  - **Housekeeping data**
  - **Monitor Rate data**
  - **Matrix data (from the classification board)**
    - **Proton and Alpha Moments**
    - **Reduced Proton and Alpha Distributions**
    - **Heavy Ion Distributions**
  - **Raw Event Data**
    - **Event Prioritization**
  - **Beacon Mode data**

Burst System Requirements

- High time resolution sampling of selected data
  - MAG @ 32Hz
  - SWEA, STE at 2 seconds
- Continuous data collection into a circulating buffer
  - Buffer sized for on the order of 10 minutes of high time resolution data
- Continuous evaluation of burst trigger criteria based on instrument data
  - Trigger on changes in SWEA count rate in a selected energy band
  - Trigger cooperatively with SWAVES based on SWAVES activity
- Freeze collection buffer a fixed interval after the trigger event to provide data both before and after the trigger
- Continue to look for more triggers,saving the “best” event in the time interval between burst playback
  - Play back burst data using a fraction of the normal telemetry on a continuous basis.

IDPU Software Development Plan

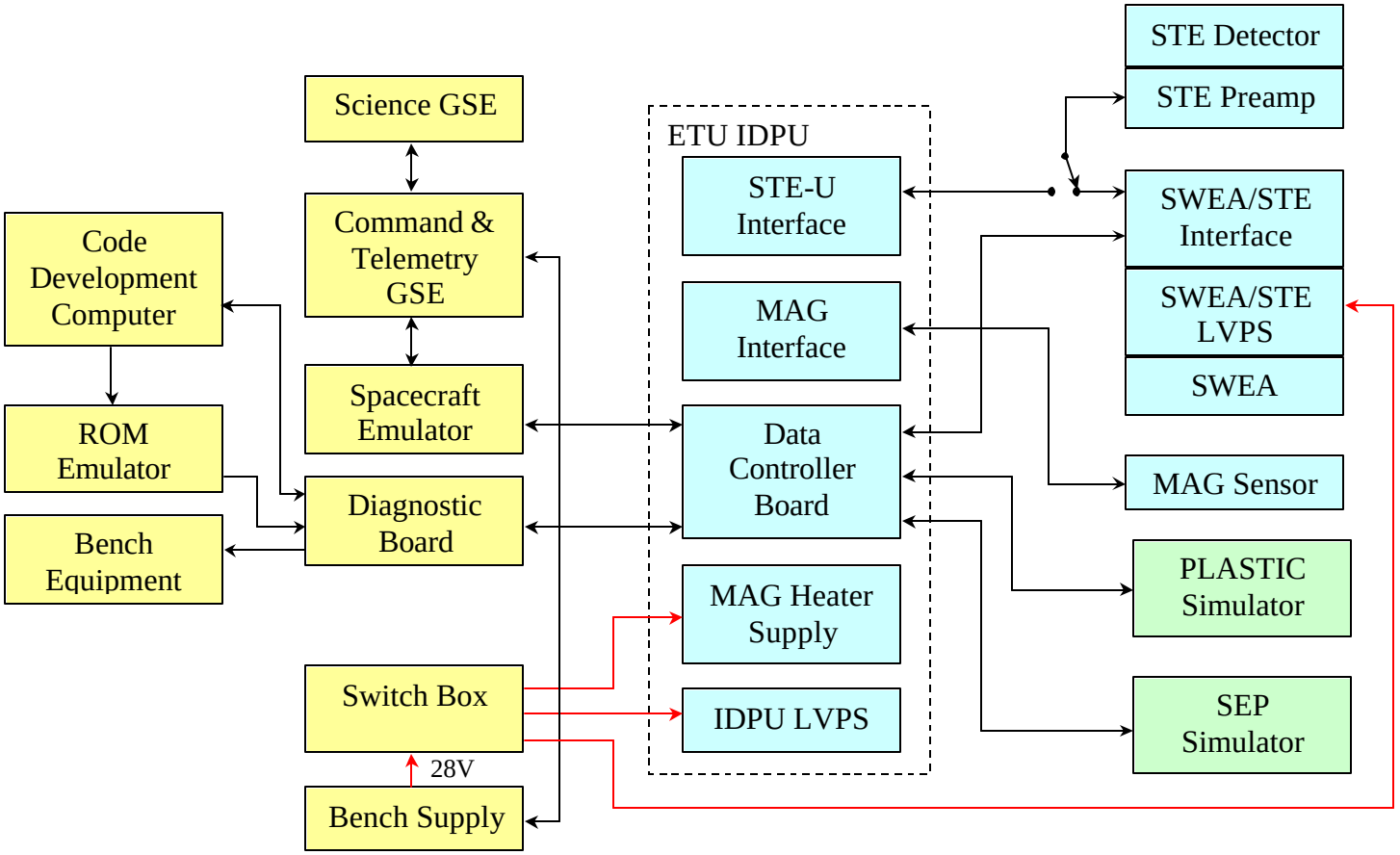
Software Development Plan

- First draft Software Development Plan 8/01 (Rev B 11/01)
- IDPU Flight Software to be developed at UCB by a two programmers
  - Most of the code is developed by a single programmer
  - PLASTIC code to be developed by a second programmer
- Code to be developed in modular, structured assembly code
- Software to be developed and tested on the ETU IDPU, with ETU Instruments and/or simulators
- Software to be tested at the module level, then at the IDPU level in an acceptance test, and finally at the system level in interface and suite tests
- Software to be developed in builds of progressing complexity:
  - First Build to test processor hardware (Complete)
  - Second build provides at least minimal functionality required to verify suite hardware (without automation and more complex telemetry products)
  - Third build should be for Flight
- Prior to installation in the flight hardware, code is maintained using a log book which contains the development history, testing, problems, etc.
- Following installation in flight hardware, software is under the same control as the hardware (configuration control, problem reports, etc.)

Software Development Tools

- Use Commercial Phyton PC-based 80C196 development tools
  - Assembler/Linker
  - Simulator (for early module testing)
- Diagnostic board connects via DCB diagnostic connector
  - Provides assorted processor bus signals to logic analyzer, LEDs
  - External PROM option, which can contain Monitor code
  - RS232 drivers for processor diagnostic serial interface (code load, diagnostics)
- Logic analyzer (for those subtle timing problems)
- ETU Data Controller Board (second in fabrication for PLASTIC)
- EGSE:
  - APL Spacecraft Emulator
    - UCB-developed Command and Telemetry GSE
    - UCB-developed Science Display GSE (MAG,SWEA,STE)
    - SEP, PLASTIC Science Display GSE
  - UCB-developed Instrument Simulator
  - Instrument ETUs
  - UNH-provided PLASTIC simulator

IDPU Software Development Setup



Software Development Schedule

• Software Development Plan	PDR (9/01) Ö
• Software Requirements Document(s)	PDR (9/01) Ö
• Software Requirements/Design Review	12/01 Ö
• Software Design Complete, start Coding	1/02 Ö
• First Build Complete, ETU Available	8/02 Ö
• IDPU Software Design Peer Review	8/02
• Software Design Review (part of CDR)	CDR (11/02)
• Critical Code Walkthrough / Acceptance Test Review	6/03
• Boot PROM Acceptance Test	8/03
• Boot PROM Install in DCB of FM1	9/03
• Second Build Complete	9/03
• Third Build Complete, Acceptance Test	3/04

IDPU Software Design

IDPU Software Architecture

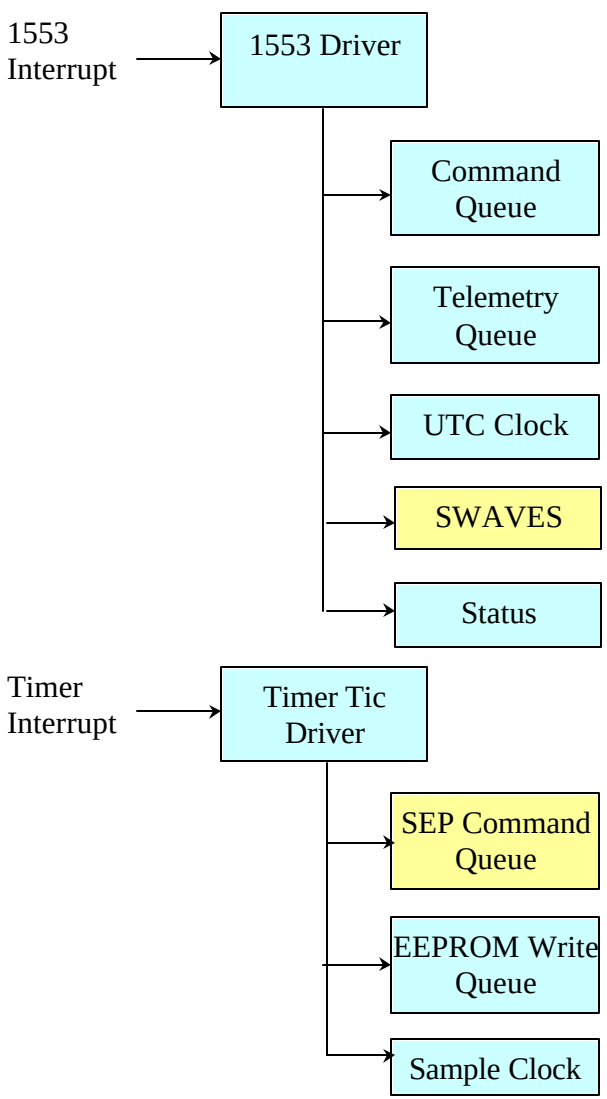
- Simple Polled-Loop operating system
  - 1553 and 256Hz interrupts take care of time-critical functions
  - Input data FIFOs hold data until the relevant task comes to collect it
  - The majority of the tasks involve short intervals of processor to deal with data, so the polling loop frequency is expected to be much greater than 1Hz, providing adequate response time to all tasks.
  - No task in polling loop requires better than 1 second latency
  - A few tasks run infrequently require more processing (such as the PLASTIC matrix data formatting) and will be broken up to level the processing load
  - A somewhat more complex priority-based time-slice operating system has been considered, but is currently more complex than is required
- Tasks are modular and as independent as possible
  - Especially PLASTIC, to allow independent operation by UNH
- Hardware designed to relieve software of most time-sensitive tasks
  - Instrument sequencers allow instruments to cycle independently
  - DMA system collects data into IDPU memory automatically
- The watchdog timer will be reset by the polling loop (timeout = 2 seconds)
  - The watchdog will monitor the interrupt tasks to verify operation
  - No 1553 activity for 10 seconds causes a 1553 interface reset.
  - No 1553 activity for 20 seconds causes a processor reset
- Software will regularly re-load all instrument registers to protect against SEU

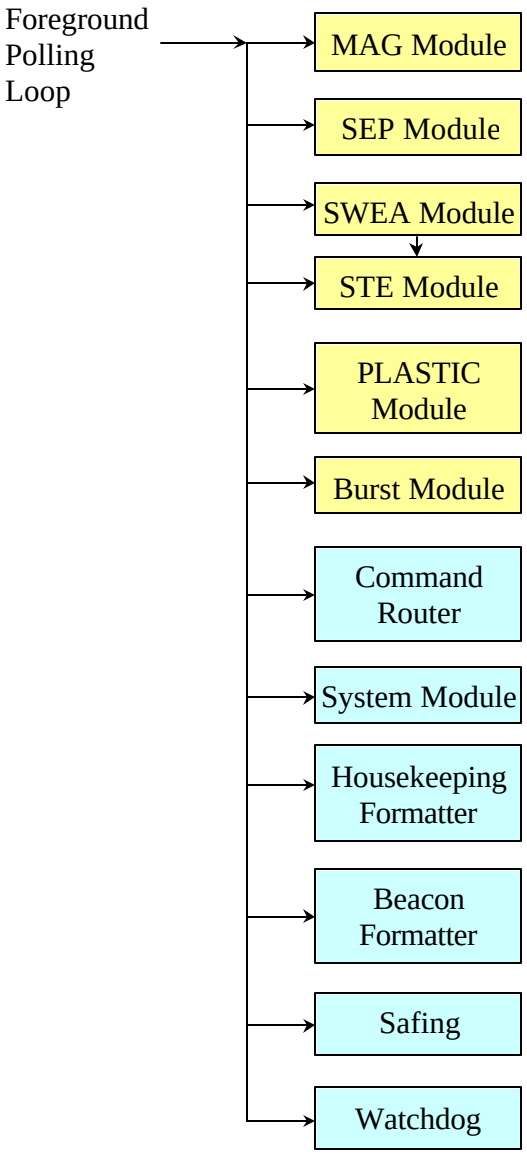
Memory Usage

- System boots on PROM
- Code is run from EEPROM nominally
  - RAM is SEU sensitive
  - Run from RAM only when writing to EEPROM
- Processor internal RAM is used for temporaries, stack
- External (3Mbyte) RAM is used for:
  - Burst memory
  - Data buffers
  - 1553 Interface
  - Parameter tables
  - Code during EEPROM write
- Parameter tables and 1553 control tables are check-summed and verified periodically for SEU
- Data SEU error rate is insignificant (less than transmission error rate)

Boot Sequence

- System boots from PROM
- PROM verifies the EEPROM code image checksum
  - On bad checksum, system waits in PROM for new code uplink from the ground via spacecraft 1553 interface
- PROM waits 10 seconds for a possible abort message prior to transferring control to EEPROM
  - During delay a subset of foreground polling loop is run to run 1553 interface, housekeeping, watchdog, etc.
- EEPROM initializes each task and then starts foreground polling loop
- EEPROM can be reloaded or patched from BOOT PROM or from a copy of the EEPROM code in RAM.





IMPACT Task Scheduling Diagram

David Curtis

1553 Handler

- **Invoked by interrupt from 1553 Interface**
  - **Completed message transfer**
- **Telemetry Message:**
  - **Fetch a new packet pointer from telemetry queue (if available) and setup 1553 for next transfer**
- **Command Message:**
  - **Queue FLTCs for foreground Command Router**
- **Timing/Status Message (1/sec):**
  - **Update UTC clock**
  - **Invoke command scripts on events (instrument power-down or thruster firing)**
- **SWAVES Interface (1/sec):**
  - **Pass SWAVES data to Burst Trigger Task**
  - **Set up to send IMPACT Burst Trigger data to SWAVES**

Telemetry Queue

- Accepts CCSDS telemetry packets from various tasks
  - MAG
  - SWEA
  - STE
  - SEP
  - PLASTIC
  - Housekeeping
  - System (memory dump)
  - Beacon
  - Burst
- Provides packets in a FIFO mode to 1553 handler
  - Implemented as a linked list. Each task controls its own packet buffers. When packet is transmitted it is marked as free by the 1553 telemetry handler so the source task can identify overrun conditions.
- Tasks are expected to generate telemetry at their allocated rate
  - Some subsystems will have programmable telemetry rates. The IMPACT Team must coordinate rate allocations between tasks.

Housekeeping & Beacon Modules

- Tasks provide housekeeping data to housekeeping formatter
  - Each task has a fixed allocation in the housekeeping packet
- Housekeeping packet rate is nominally once a minute
  - Can be increased during I&T and commissioning to improve test sequence speed
- Tasks provide Beacon data to Beacon formatter
  - Each task has a fixed allocation in the housekeeping packet
- Beacon Packets generated once a minute each for PLASTIC and IMPACT

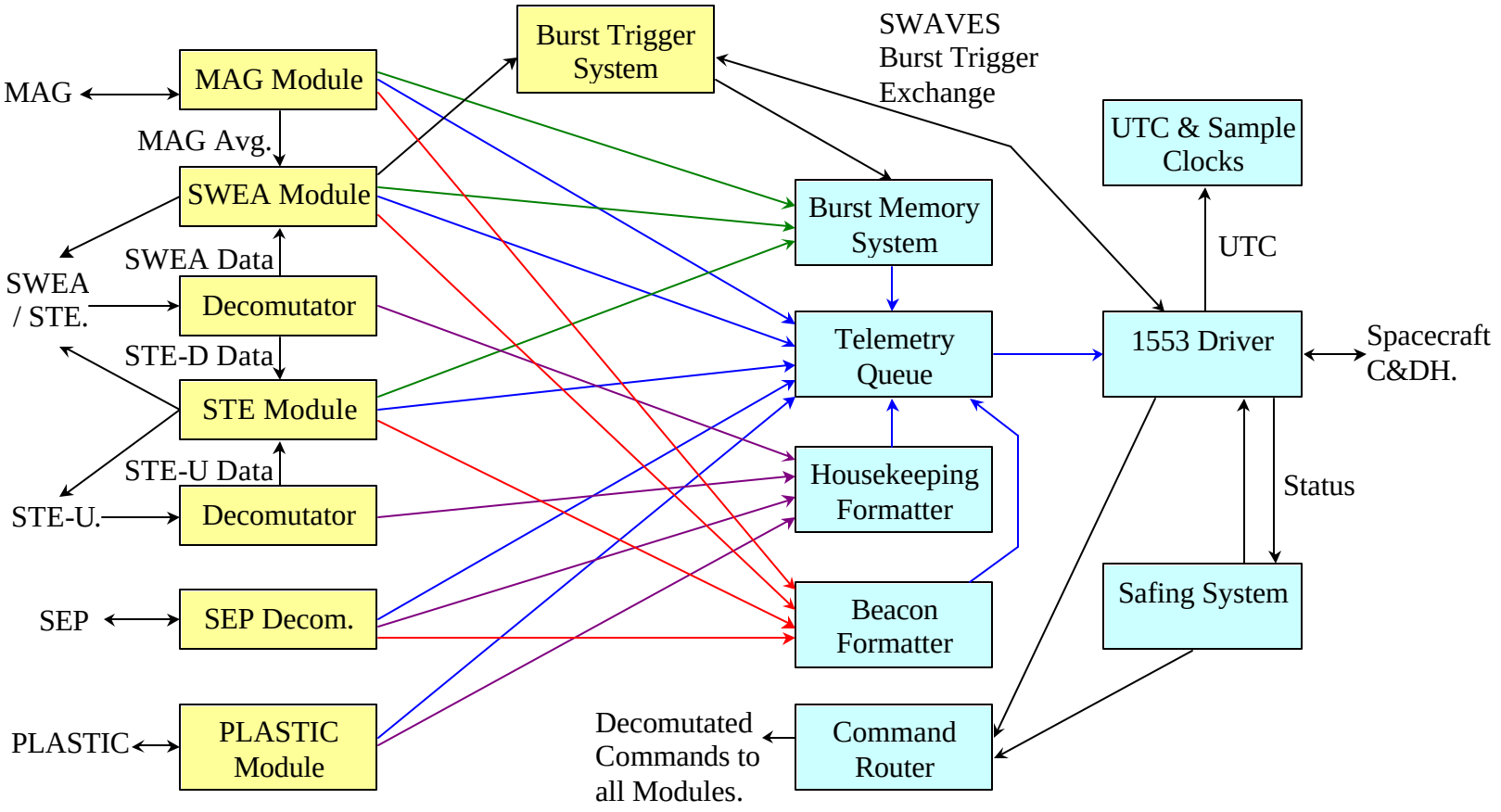
Safing System

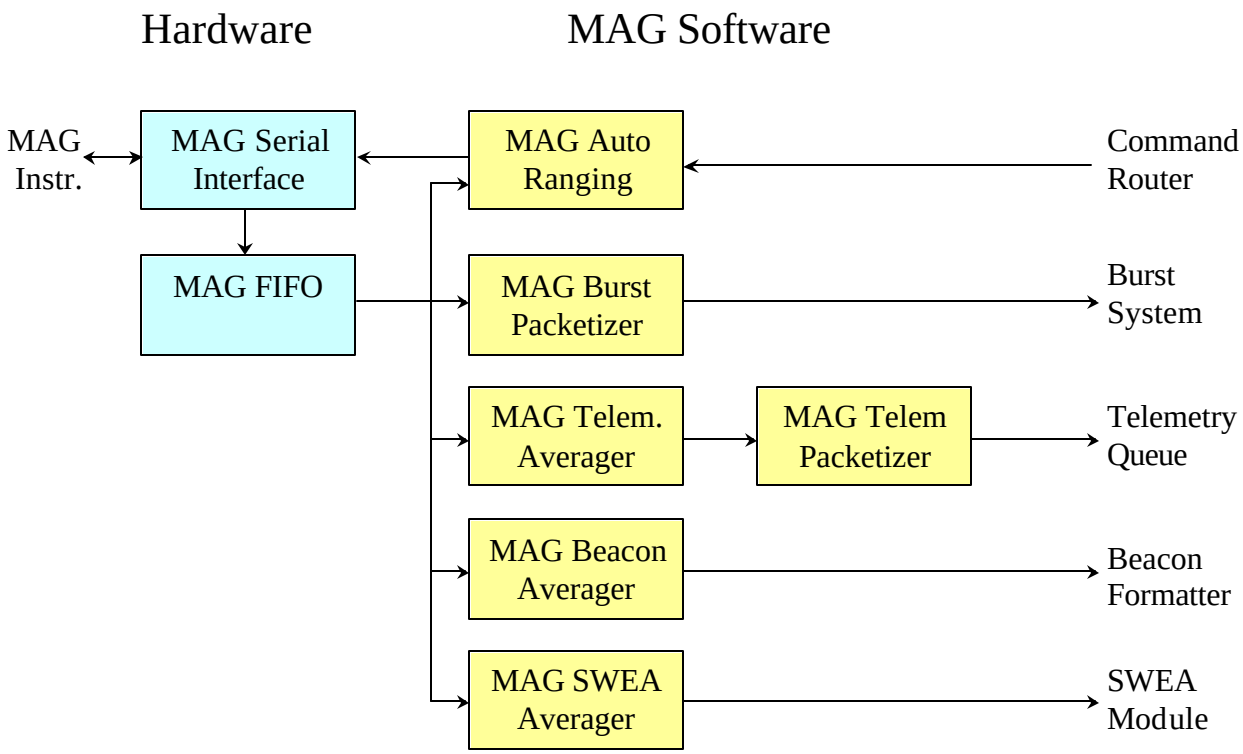
- This system is used to invoke a sequence of operations to be performed in response to an event
- The operations are executed at a fixed rate (one command a second, with an N second delay capability)
- Operations are in the form of command packets which are distributed just as if they were received over the 1553 interface
- Operation sequences are programmable from the ground to allow flexibility
- Operation Sequences can be invoked by tasks or by command
- Currently planned sequences:
  - Power Down Preparation (1553 warning)
  - Thruster Firing Preparation (1553 Warning)
  - End of Thruster Firing (1553 Warning)
  - Fine Pointing State (1553 Warning)
  - PLASTIC HV Discharge
  - Temperature / Voltage / Current limits

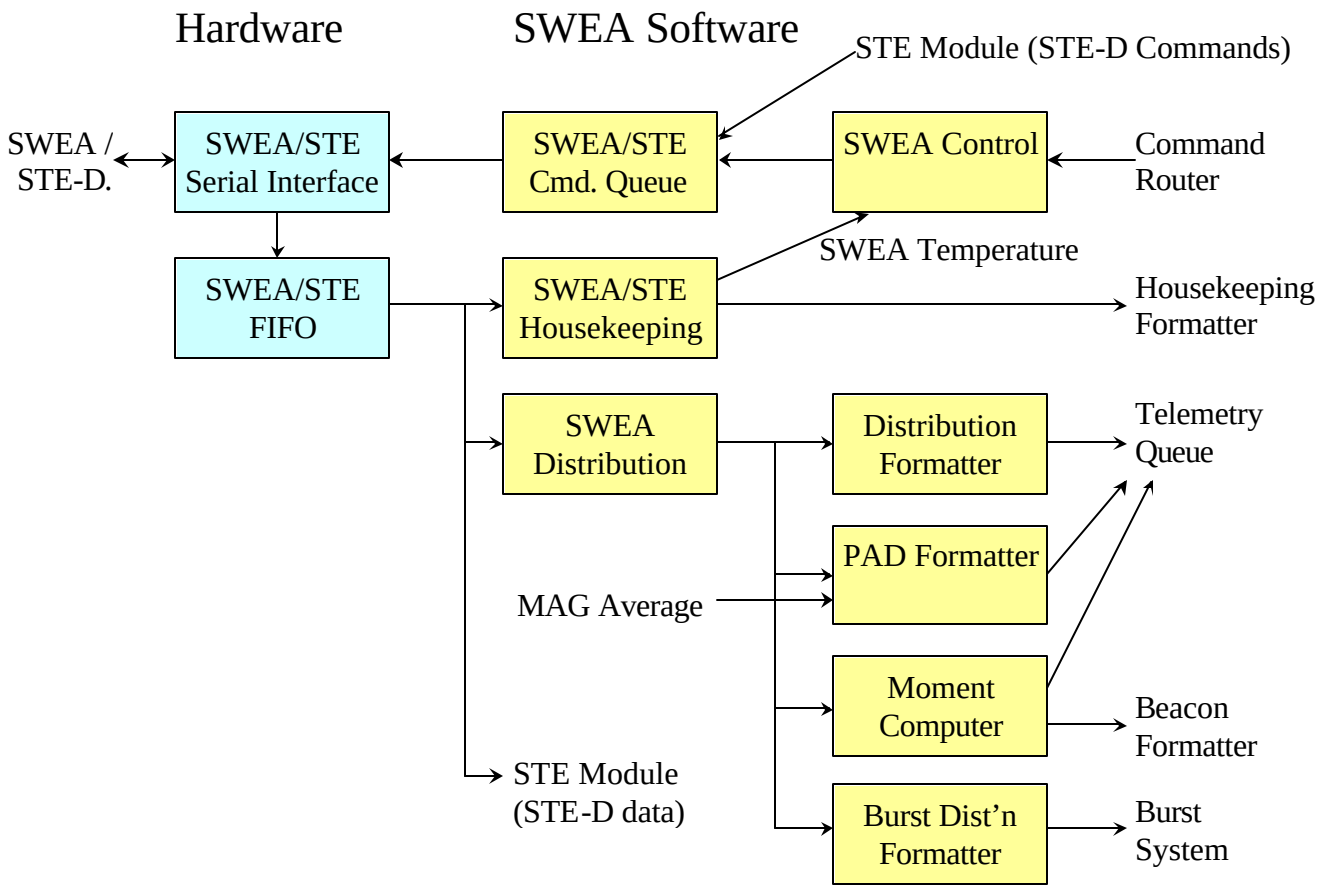
Burst System

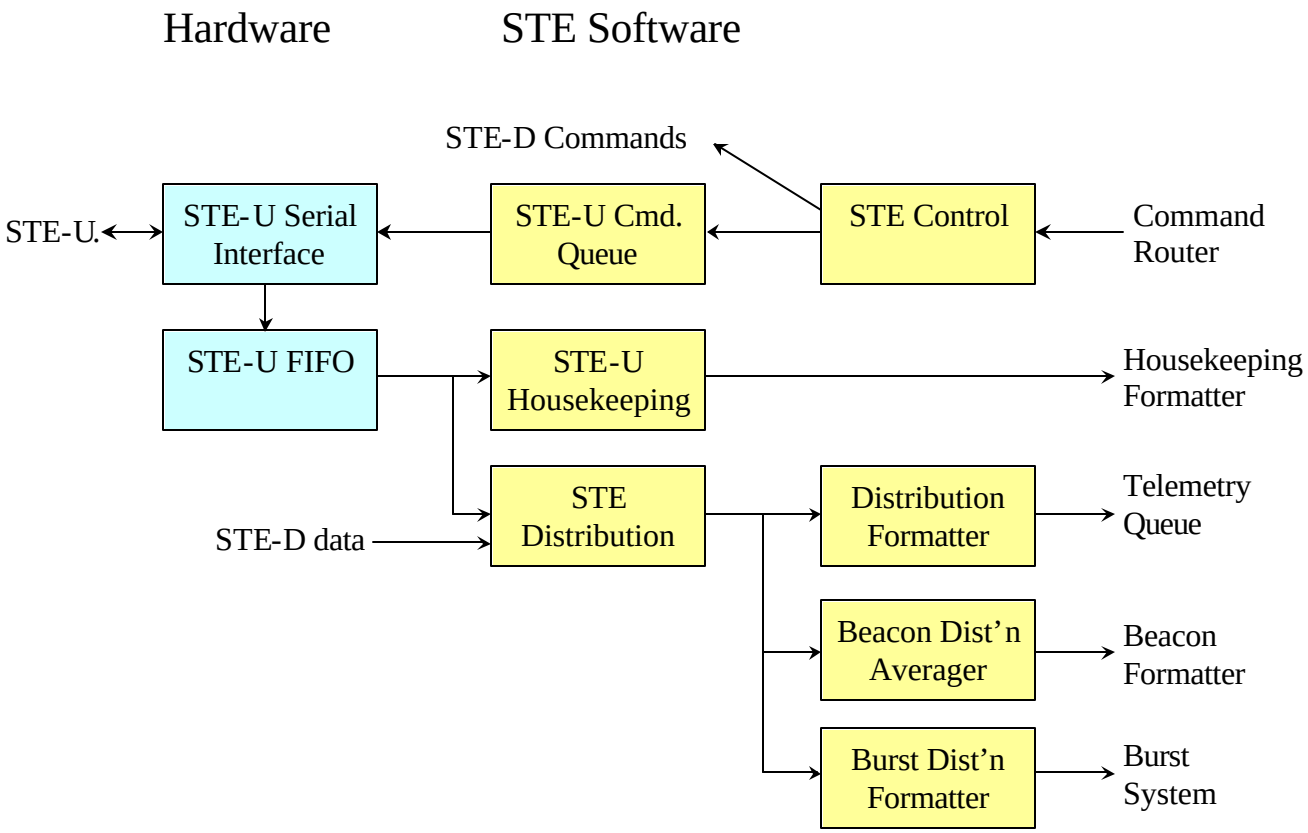
- Tasks provide a continuous stream of packets into the Burst system
  - MAG, SWEA, STE
- The Burst system records the data into a circular buffer in RAM
  - This allows pre-trigger data to be saved
  - Three buffers are allocated in RAM: Transmitting, Best, and Current
- A trigger criteria system evaluates data from SWEA and SWAVES to detect an event
- If a new event has higher criteria than the previously recorded "Best" event, the current event collection is completed, and the new buffer is saved as "Best"
  - The old "Best" buffer now becomes "Current" and is over-written with the continuous telemetry stream, looking for an even better burst
- This continues until the "Transmitting" buffer has been read out to telemetry. The "Best" buffer now becomes "Transmitting", and the old "Transmitting" becomes "Current".

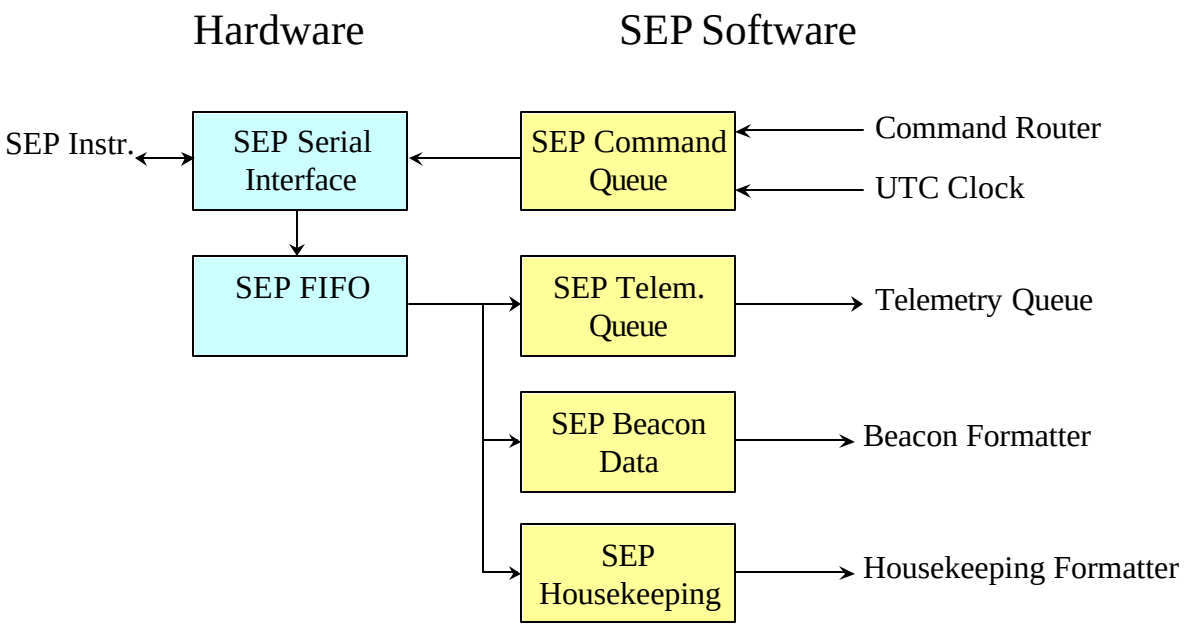
IMPACT Data Flow Diagram

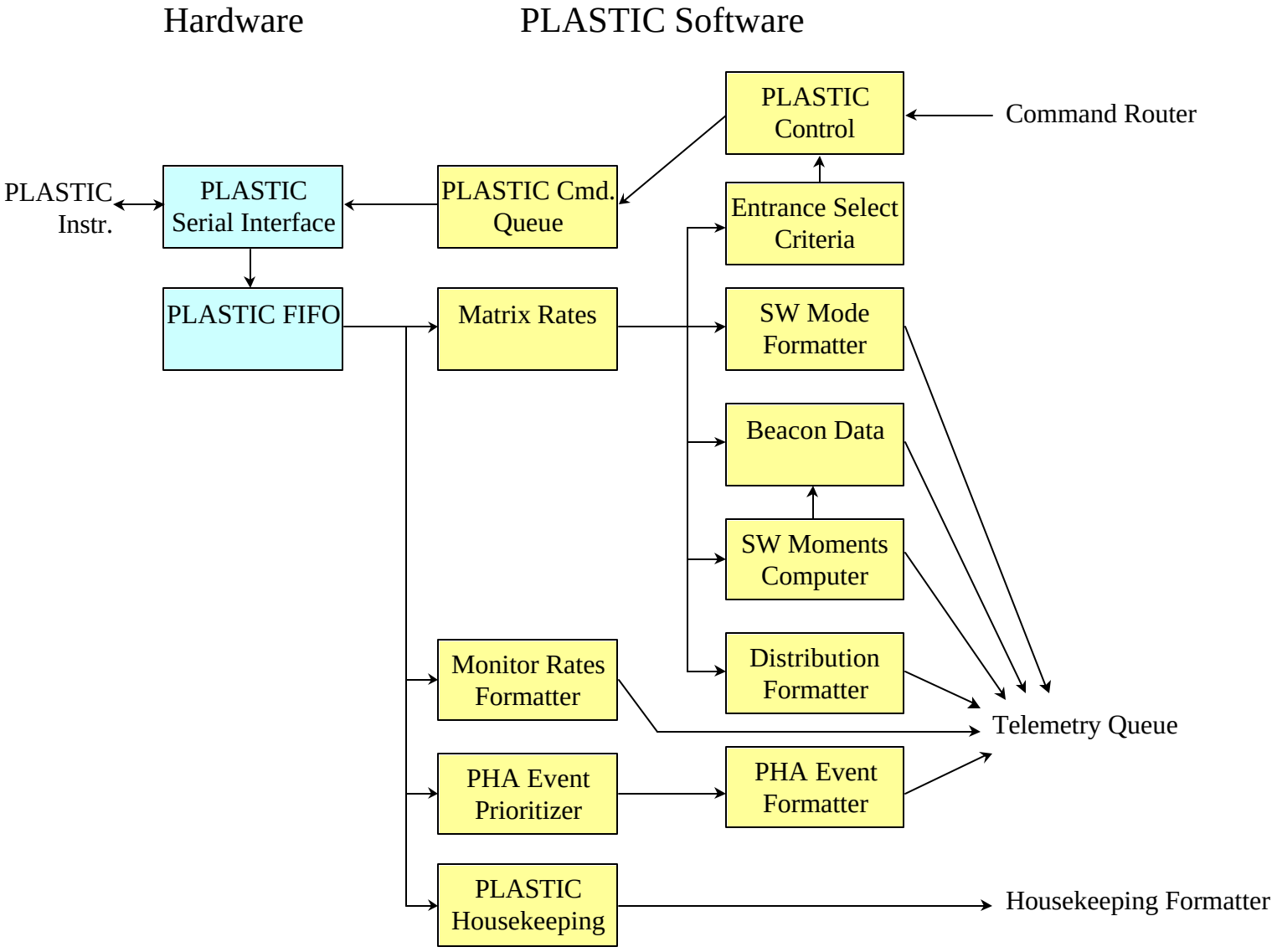



Hardware
MAG Software
MAG Instr.
MAG Serial Interface
MAG FIFO
MAG Auto Ranging
MAG Burst Packetizer
MAG Telem. Averager
MAG Telem Packetizer
MAG Beacon Averager
MAG SWEA Averager
Command Router
Burst System
Telemetry Queue
Beacon Formatter
SWEA Module


Hardware
SWEA Software
STE Module (STE-D Commands)
SWEA /
STE-D.
SWEA/STE
Serial Interface
SWEA/STE
Cmd. Queue
SWEA Control
Command
Router
SWEA Temperature
SWEA/STE
FIFO
SWEA/STE
Housekeeping
Housekeeping
Formatter
SWEA
Distribution
Distribution
Formatter
Telemetry
Queue
PAD Formatter
MAG Average
Moment
Computer
Beacon
Formatter
STE Module
(STE-D data)
Burst Dist'n
Formatter
Burst
System

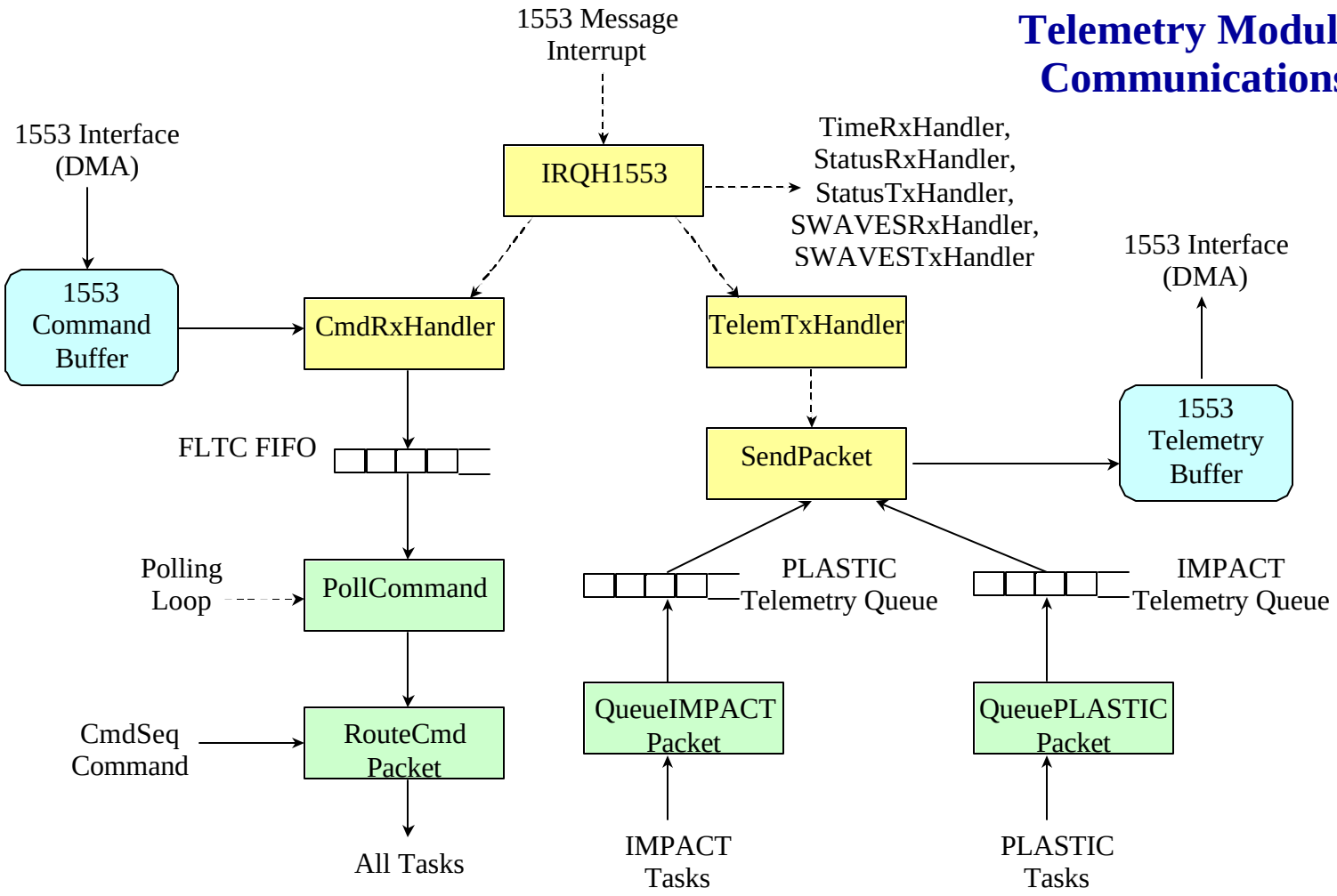

Hardware
STE Software
STE-D Commands
STE-U.
STE-U Serial Interface
STE-U Cmd. Queue
STE Control
Command Router
STE-U FIFO
STE-U Housekeeping
Housekeeping Formatter
STE-D data
STE Distribution
Distribution Formatter
Telemetry Queue
Beacon Dist'n Averager
Beacon Formatter
Burst Dist'n Formatter
Burst System


Hardware
SEP Software
SEP Instr.
SEP Serial Interface
SEP Command Queue
Command Router
UTC Clock
SEP FIFO
SEP Telem. Queue
Telemetry Queue
SEP Beacon Data
Beacon Formatter
SEP Housekeeping
Housekeeping Formatter

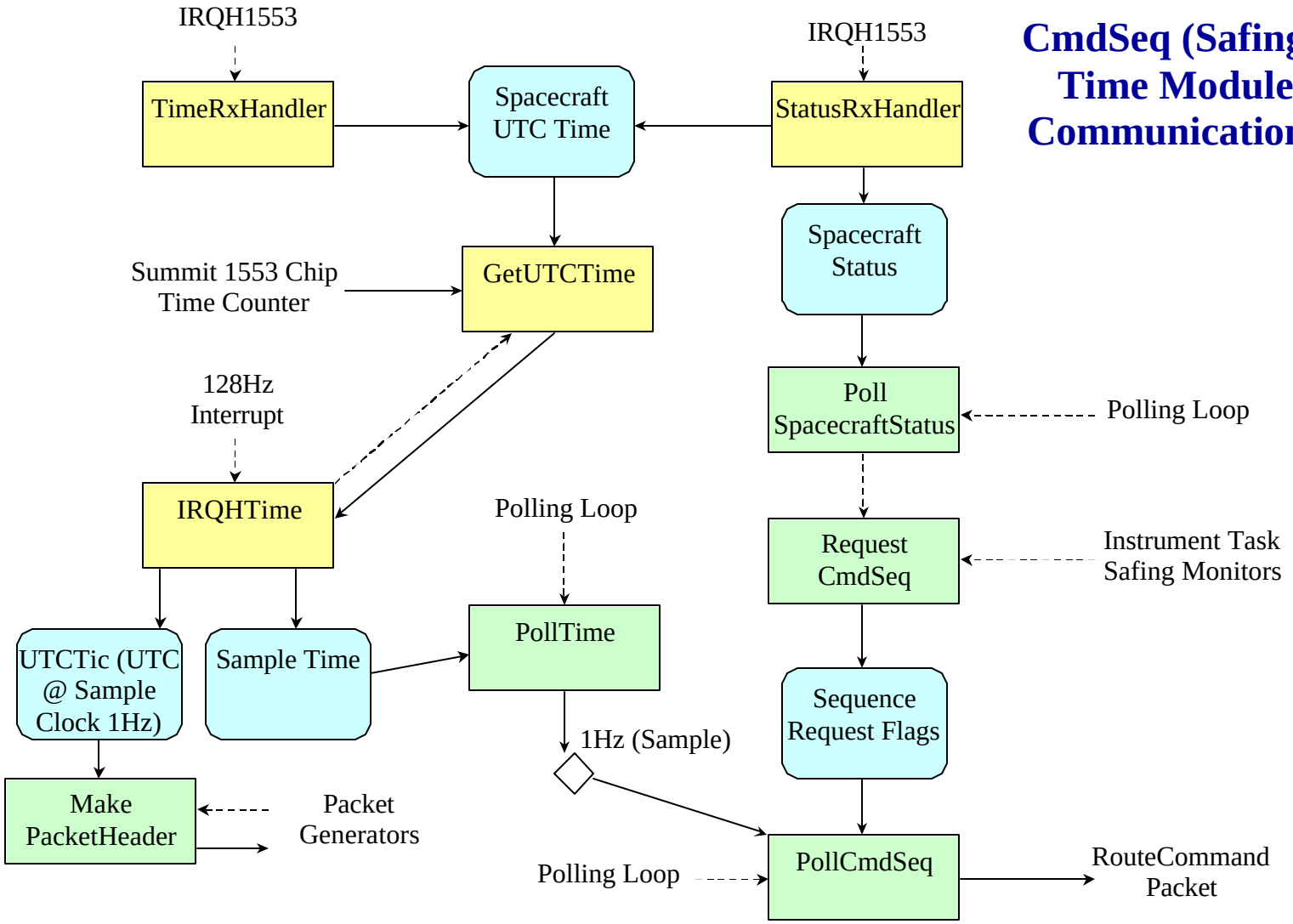

Hardware
PLASTIC Software
PLASTIC Instr.
PLASTIC Serial Interface
PLASTIC FIFO
PLASTIC Cmd. Queue
PLASTIC Control
Command Router
Entrance Select Criteria
Matrix Rates
SW Mode Formatter
Beacon Data
SW Moments Computer
Distribution Formatter
Monitor Rates Formatter
Telemetry Queue
PHA Event Prioritizer
PHA Event Formatter
PLASTIC Housekeeping
Housekeeping Formatter

Detailed Task Communications Diagrams

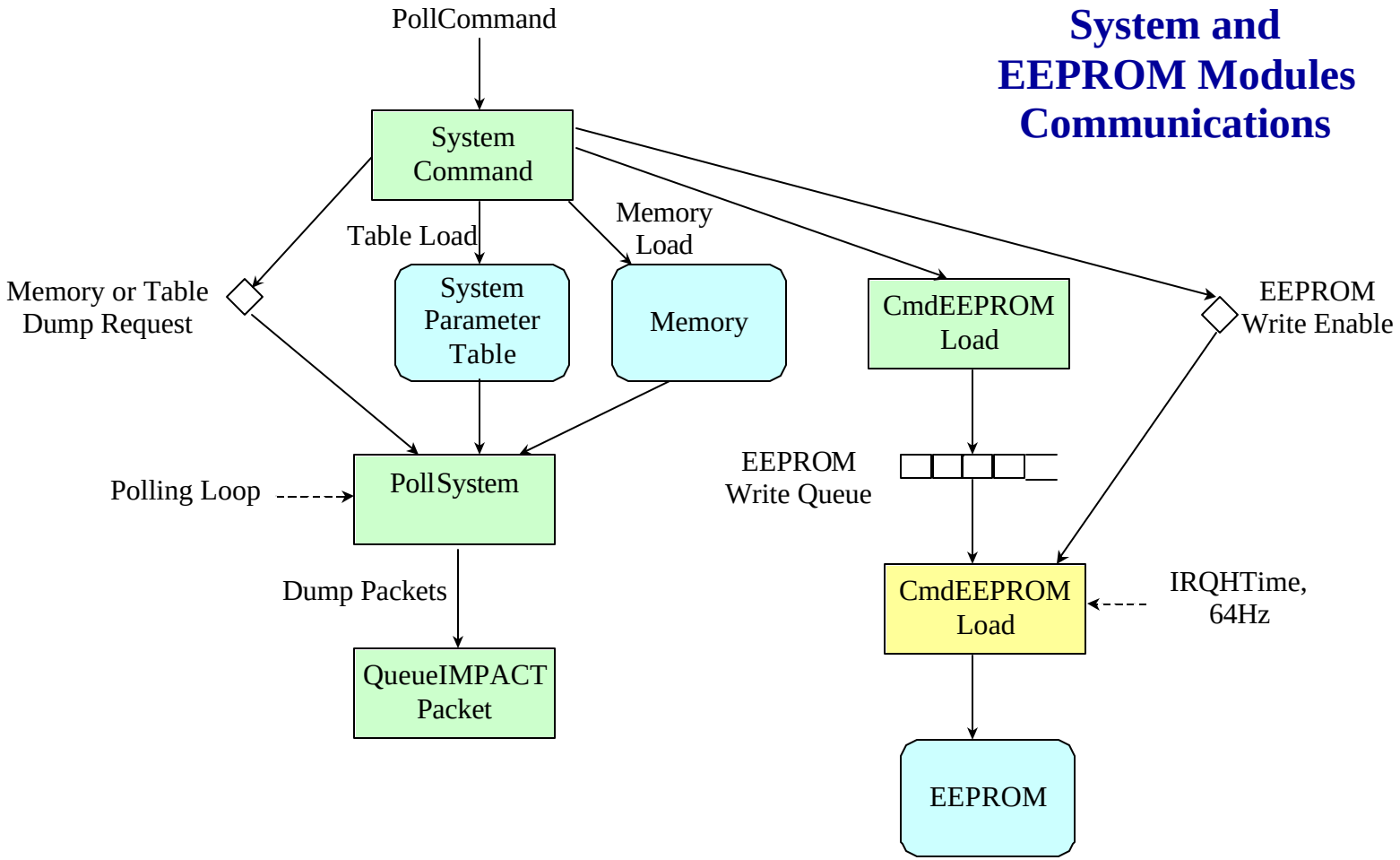
Command & Telemetry Modules Communications



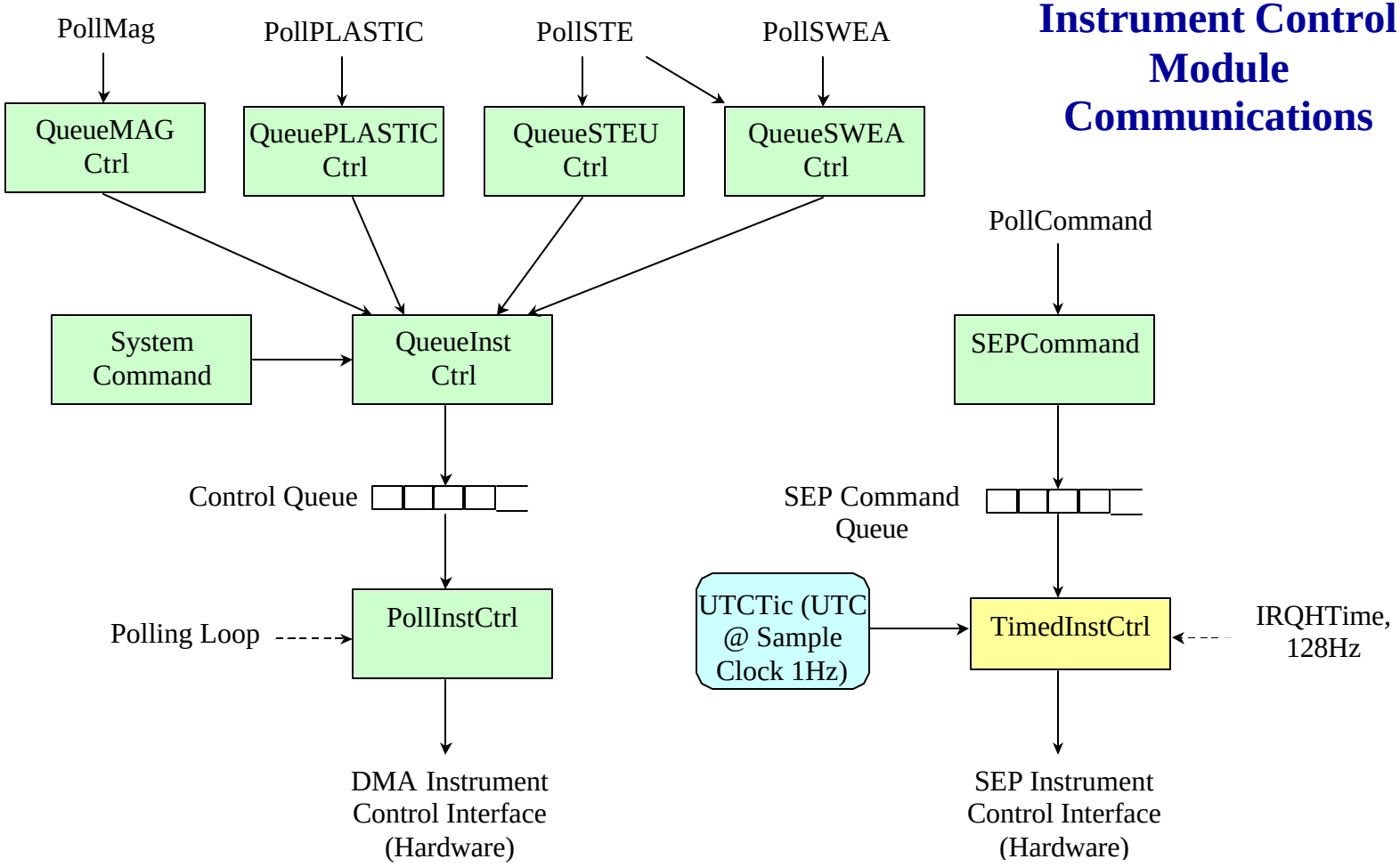
CmdSeq (Safing), Time Module Communications



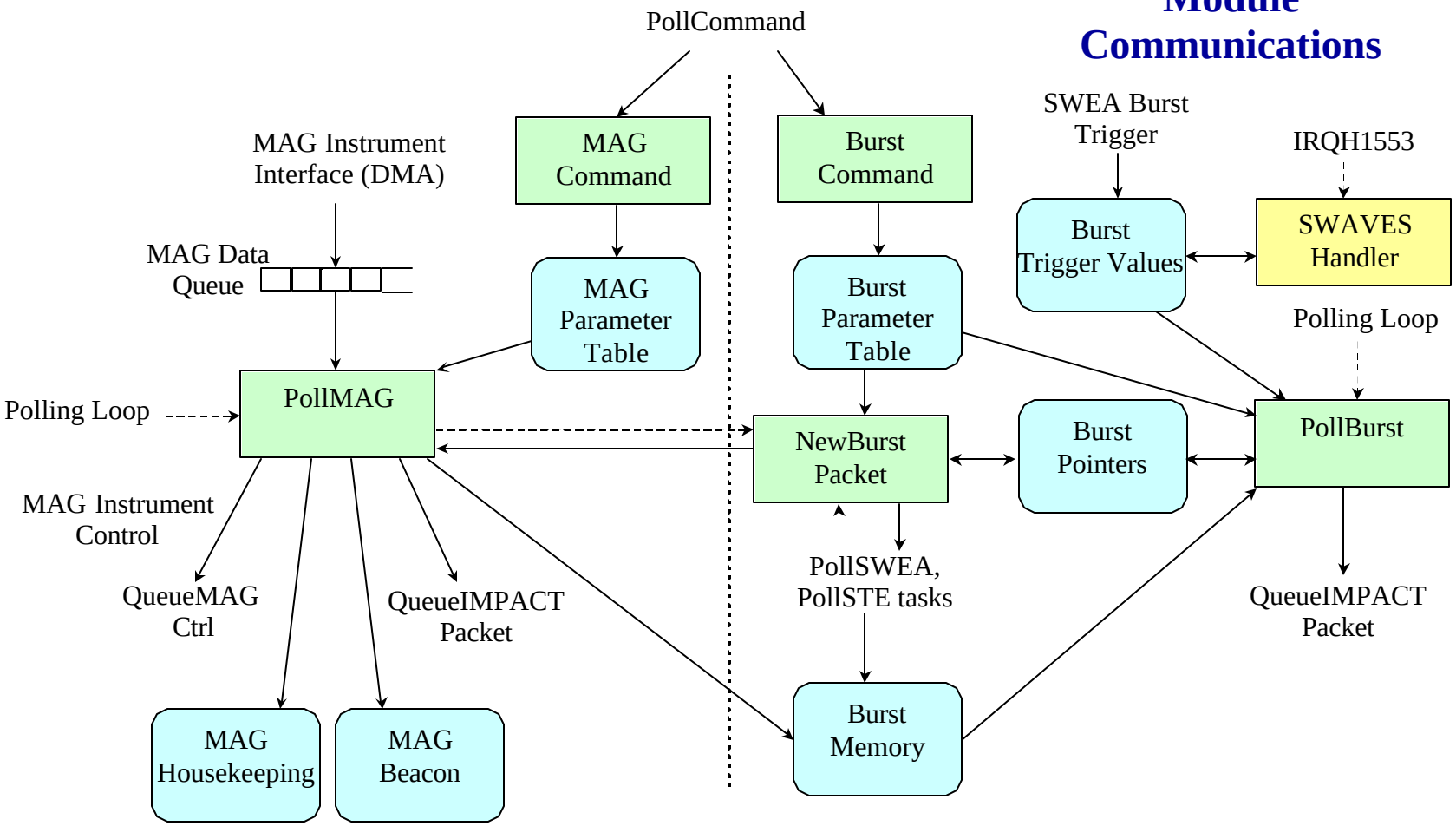
System and EEPROM Modules Communications



Instrument Control Module Communications



Mag and Burst Module Communications



IDPU Flight Software Resource Estimates

Task	Processor Cycles, %	Code Size, kbytes	Buffer Memory, kbytes
Service Software:			
System	1%	1.3 (b)	66.3
1553 Interface	1%	1.7 (b)	2.0
Command Router	<1%	0.5 (b)	2.0
Telemetry Packet Queue	<1%	0.2 (b)	0.1
Burst System	1%	1.0	1500.0 (*)
Time	<1%	0.3 (b)	0.1
Housekeeping	<1%	0.2 (b)	0.8
Beacon Telemetry	<1%	0.2	0.6
EEPROM Manager	<1%	1.1 (b)	2.0
Misc	<1%	0.4 (b)	0.3
MAG Software	1%	1.3	1.2
SEP Software	1%	0.8	3.1
SWEA Software:			
SWEA Moments	20%	1.0	17.0
SWEA Distributions	4%	2.5	10.8
SWEA Misc.	1%	1.0	2.5
STE Software	1%	1.0	3.7
PLASTIC Software:			
PLASTIC Distribution	6%	2.0	185.6
PLASTIC PHA	5%	0.5	46.6
PLASTIC Moments	5%	1.0	45.6
PLASTIC Misc.	1%	1.0	2.5
Total	**55%**	**19.0**	**1892.8**
Available	100%	64	3072
% Usage	55%	29%	62%

(*) Nominal value. Burst memory will expand to use all unallocated memory space.

(b) To be included (at least in part) in Boot PROM. PROM = 8kbytes, usage = 72%

Software Status

- **Build #1 is complete**
  - **Written by the IDPU hardware designer to verify the hardware**
- **Build # 2 is in progress**
  - **A lot of code was written while waiting for the DCB ETU to be available**
  - **DCB ETU was delivered recently and code debugging has started.**
  - **Boot PROM code runs, generates housekeeping**
- **The following modules have been written:**
  - **Boot**
  - **System**
  - **1553**
  - **Time**
  - **Command Router**
  - **Command Sequencer**
  - **Telemetry**
  - **Status**
  - **Housekeeping**
  - **Watchdog**
  - **EEPROM**
  - **RS232 (diagnostic)**
  - **Instrument Controller**
  - **MAG**
  - **SEP**
  - **STE (in progress)**

Software Status (continued)

- **Modules remaining to write for Build 2:**
  - **SWEA (no moments)**
  - **PLASTIC (simple)**
- **Code for Build 3**
  - **Burst Trigger (including SWAVES exchange)**
  - **Beacon**
  - **SWEA Moments**
  - **PLASTIC Moments**
  - **PLASTIC Solar Wind Tracking**
  - **Perhaps some of the other more complex PLASTIC code, such as event prioritization.**

Module Headers

```
;  BOOT.MCA
;  Stereo IMPACT IDPU Flight Software
;  Boot Module
;  - This module is the PROM boot code.
;        Note that on RESET, the code segment points
;        to the PROM memory, so code is executed
;        out of PROM
;  - On reset the processor jumps to START
;  - The processor is initialized (stack, etc.)
;  - The PROM code is copied to the RAM code segment,
;    the code segment is changed to RAM, and the PROM
;    is turned off to save power and provide standard
;    code timing (PROM access is slow because it is
;    8-bits wide)
;  - PROM tasks (including Timer & 1553 interrupts,
;    but excluding the instrument interfaces) are
;    initialized
;  - The foreground polling loop of PROM tasks
;    is started
;  - Polling continues until a timeout, then the code
;    segment is transferred to a selected EEPROM code
;    image (if the EEPROM image is valid)
;        Default EEPROM code image is zero, but can be
;        set by command during the timeout interval.
;  - If EPROM image is bad or a command requests PROM
;    code, the system remains in the PROM code image in
;    RAM.
```

```
;  DRV1553.MCA
;  Stereo IMPACT IDPU Flight Software
;  1553 Driver for the UTMC Summit 1553 chip
;  This module contains:
;    Init1553:  Initializes the Summit chip and 1553
;      interrupt handler.
;    IRQH1553:  the 1553 interrupt handler.  This is
;      invoked by the Summit chip in case of an access
;      to one of the 1553 message buffers, or an error
;      The handler determines what if any buffers were
;      accessed and calls an appropriate handler for that
;      message (see message handler list below).
;    SendPacket: copy a telemetry packet to the 1553
;      message buffers used by the spacecraft to collect
;      telemetry.  Called by TelemTxHandler, which is
;      called by IRQH1553 when the previous packet has
;      been transferred.
;    Poll1553DCB: check the contents of the Summit DCB
;      table in RAM.  This table is used by the Summit
;      chip to determine what to do with message traffic.
;      Called by the foreground polling loop.
```

```
;  Command.MCA
;  Stereo IMPACT IDPU Flight Software
;  Command packet receiver, decoder, and router
;  Includes:
;    InitCommand - initialize FLTC queue
;    CmdRxHandler - called from 1553 interrupt handler
;      when there is a new command FLTC message.  Queue
;      the message in the FLTC buffer for later processing.
;    PollCommand - Polled by foreground system.  Read any
;      FLTCs from the queue and decode them (ParseFLTC)
;      into CCSDS command packets.  Then decode
;      (DecodePacket) and route (RoutePacket) via
;      the command router table (CommandHandlerTable)
;      Note that PROM and EEPROM have different
;      CommandHandlerTables since EEPROM has more modules
```

```
;  SWAVES.MCA (STUB)
;  Stereo IMPACT IDPU Flight Software
;  SWAVES/IMPACT Status Exchange handler
;  Called from 1553 handler.  Includes:
;    SWAVESRxHandler:  Deal with SWAVES message received
;    SWAVESTxHandler:  Deal with SWAVES ready for
;        another message
```

```
;  STATUS.MCA
;  Stereo IMPACT IDPU Flight Software
;  Spacecraft/Instrument Status Exchange handler
;  Called from 1553 handler.  Includes:
;    StatusRxHandler: deal with UTC time code and
;      spacecraft status information passed once a second
;      to the instrument via 1553
;    PollSpacecraftStatus:  called from foreground processing
;      loop.  Monitor status changes & run appropriate
;      response command sequences
;    TimeRxHandler: called by DRV1553 when Time Tic message
;      received (at this time the time counter in the
;      Summit 1553 chip is re-loaded with the spacecraft time
;      sub-seconds counter
;    StatusTxHandler:  send instrument status information
;      to the spacecraft via 1553.  Includes power service
;      shutdown requests, built up from individual
;      instrument shutdown calls.  Note that SYSTEM
;      includes command to request shutdown so they can be
;      incorporated into event response command scripts
;    ShutdownIDPU: request IDPU power shutdown
;    ShutdownPLASTIC: request PLASTIC power shutdown
;    ShutdownSSEP: request SEP power shutdown
;    ShutdownSWEASTE: request SWEA/STE-D power shutdown
```

```
; TELEM.MCA
; Stereo IMPACT IDPU Flight Software
; Telemetry Queue handler.  Maintain a queue of telemetry
;   packets to be sent.  Queue is maintained as a linked list to
;   avoid moving data around.  Separate queues for IMPACT and
;    PLASTIC are maintained since IMPACT instruments tend to
;   make a few packets fairly frequently, while PLASTIC makes
;   many packets infrequently. Separate queues allows IMPACT
;   to avoid large numbers of packets stuck behind blocks of
;   PLASTIC packets.
; Note that there is no reason why an occasional PLASTIC packet
;   could not be sent via the IMPACT queue, or a large IMPACT
;   block of packets could not be sent via the PLASTIC packet
;   queue.
; Includes:
;   TelemTxHandler:  Called by 1553 Interrupt handler when a
;      fresh telemetry packet is required.  Pick the next packet from
;      the queue and install into the 1553 buffer using the
;      SendPacket call.
;   QueueIMPACTPacket:  Add a new telemetry packet to the
;      IMPACT packet queue.
;   QueuePLASTICPacket:  Add a new telemetry packet to the
;     PLASTIC packet queue.
```

```
; TIME.MCA
; Stereo IMPACT IDPU Flight Software
; Timing routines including Timer Tic Interrupt Handler:
;   InitTime:  Initialize time counters, Timer Tic interrupt
;   PollTime:  Called from foreground polling loop.  Measure
;      maximum time between calls (sent in HSK packet) to verify
;      system is keeping up.  Also set a flag when a new second
;      tic has occurred, for foreground routines that only need
;      service once a second or less.
;   GetSampleTime:  Get the current value of the sample clock
;      to 1us accuracy, 15 hours dynamic range
;   GetUTCTime:  Get the current UTC time code, to 15us
;      accuracy, 136 year dynamic range (spacecraft UTC time
;      code)
;   IRQHTime: Timer Tic interrupt handler.  Run at 128Hz,
;      synchronous with the sample clock.  Maintain the sample
;      clock HH:MM:SS register sent to the instruments for
;      synchronization once a second.  Sample the UTC clock
;      at the time of the Sample Clock 1Hz tic so we can
;      accurately generate data collection time packet headers
;      (this time is also sent to SEP for its packet headers)
;      At 64Hz, poll the EEPROM write routine.  EEPROM write
;      must be synchronous to ensure we can keep up with the
;      worst case write rate, which occurs when there are a series
;      of short commands (@2000bps), each of which changes 1
;      byte in the EEPROM, and each of which takes a full
;      EEPROM write cycle to implement.
```

```
;  SYSTEM.MCA
;  Stereo IMPACT IDPU Flight Software
;  System level tasks and utilities, including:
;    InitSystem: Initialize system constants, variables, and queues
;    PollSystem: Run from foreground polling loop.  Look for
;       memory or Table dump packets to build, check the system
;       parameter table checksum, call a commanded polled task if
;       selected.
;    MakePacketHeader: Build a CCSDS telemetry packet header.
;       Called from all tasks that make telemetry packets.
;    ClrPacket:  Clear the contents of a telemetry packet.  Called
;        during initialization tasks for tasks that make packets.
;    UpdateSystemTableChecksum:  Fix the system parameter
;       table checksum.  Usually called after a command changes a
;       system parameter table value
;    UpdateTableChecksum:  Generic parameter table checksum
;       update
;    PollTableChecksum:  Generic parameter table checksum
;        tester. Called from task foreground loops.
;    SystemCommand:  Decode and execute a command to an
;       ApID in the system range (called from COMMAND in the
;       foreground).   System Commands include memory &
;       EEPROM load
```

```
;  Watchdog.MCA
;  Stereo IMPACT IDPU Flight Software
;  Watchdog code
;
;  Manages the watchdog timer, using the UT80CRH196KD
;  watchdog timer hardware.  Code includes:
;    InitWatchdog: setup and enable the hardware watchdog
;         timer (times out in 2 seconds)
;    PollWatchdog:  verify 1553 & timer interrupts active,
;         then pet watchdog.  Called from foreground loop
;    Require some timer interrupt handler activity
;       to pet watchdog
;    Use a 10-second timeout on 1553 activity.
;       If no 1553 activity for 10 seconds,
;            re-initialize 1553 system
;       If no 1553 activity for 20 seconds,
;            don't pet watchdog (causes a reset)
```

```
;  HKP.MCA
;  Stereo IMPACT IDPU Flight Software
;  Housekeeping packet generator
;  Includes:
;    InitHKP:  Initialize the housekeeping system
;    PollHKP:  Called from foreground polling loop.
;       check if it is time to generate a housekeeping packet.
;      if so build one & send it to the IMPACT telemetry queue.
```

```
;  EEPROM.MCA
;  Stereo IMPACT IDPU Flight Software
;  EEPROM Write facility
;  (also Code Page Functions)
;
;  Foreground calls CmdEEPROMLoad
;     (via Command & System modules)
;     enters a EEPROM write request into a EEPROM Write queue
;    (EEPROMWriteEnable command must be sent first)
;  TIMER calls PollEEPROMWrite under interrupt handler
;     (to maintain deterministic timing)
;     Writes up to one EEPROM block (256 bytes) to EEPROM
;  Uses about 20% of the processor time while writing
;     about 3ms each timer tic @ 64Hz
;  Max throughput = 32 blocks/second
;     (use every other cycle for verify)
;     Blocks can be up to 256 bytes (block address aligned)
;    PollEEPROMWrite breaks up commanded writes into blocks
;  Max input rate: Assuming 5 FLTC's a second with
;     up to 6 short commands each makes 30 blocks/sec
;  For long writes, 5 FLTCs/second is 310 bytes/second
;     a little over 1 block/second.
;
;
```

```
; Other entries:
;  InitEEPROMWriteQueue:  Called by InitSystem, to initialize
;     the EEPROM Write Queue
;  EEPROMWriteEnable:  Enable EPROM Write in the FPGA
;  EEPROMWriteDisable:  Disable EPROM writes in the FPGA
;    These are called by System Commands only.  The ground
;    manages this to avoid accidental EEPROM writes.
;  UpdateCodePageChecksum:  Fix a selected code page checksum.
;    including EEPROM code pages, by System Command.
;    EEPROMWriteEnable command must be sent first.
;    Note there are 5 code pages: 4 EEPROM and one RAM.
;  CheckCodePageChecksum:  Verify a selected code page
;    checksum.  Called before executing a code page by Boot.
;  ChangeCodePage:  Skip to a different code page.  The
;    code page checksum is not checked.  The new code
;    page must be nearly identical to the old one or this
;    will cause a crash.  Called by System Command.
;  CopyCodePageToRAM:  Used to copy an EEPROM code page
;    to the RAM code page.  This allows the RAM to be used
;    instead of EEPROM (required if you are going to write
;    to EEPROM, since you cannot execute out of EEPROM while
;    it is being written to).  Called by System Command.
;  InitCodeChecksum: Initialize the polled code page checksum
;    monitor
;  PollCodeChecksum:  Called from foreground routine.  Slowly
;    compute the checksum of each of the 5 code pages and report
;    in the housekeeping packet
```

```
; InstCtrl.MCA
; Stereo IMPACT IDPU Flight Software
; Instrument Control Queue handler
; Deal with the instrument command DMA system
; This includes:
;  InitInstCtrl:  Setup the instrument control system
;  TimedInstCtrl:  Deal with the non-DMA instrument
;     control system (runs together with the DMA system)
;     This system is used to trickle out SEP commands so
;     as to not flood the SEP processor.  Runs at 128Hz
;     out of the TimerTic interrupt to keep up with the
;     worst case SEP command bandwidth.
;  PollInstCtrl:  Called from foreground polling loop.
;     If the DMA queue has completed transmission and there
;     are new controls to send to instruments, swap buffers
;     and re-start the DMA.  Commands are queued by:
;  QueueInstCtrl;  add an instrument control to the DMA queue
;  QueueMagCtrl:  Same, but specific to MAG instrument control
;  QueuePLASTICCtrl:  Same, but specific to PLASTIC
;     instrument
;  QueueSWEASTECtrl:  Same, but specific to SWEA/STE-D
;     instrument
;  QueueSTEUCtrl:  Same, but specific to STE-U instrument
```

```
; CmdSeq.MCA
; Stereo IMPACT IDPU Flight Software
; Command Sequence Handler
; Runs a sequence of commands in response to some event
; Includes:
;  InitCmdSeq:  Initialize the command sequence system
;  PollCmdSeq:  Called from foreground polling loop.
;     Once a second route a command from the sequence table
;     if one is active.
;  DelayCmdSeq: Delay processing command sequence for N
;     seconds
;  RequestCmdSeq:  Request one of 32 possible command
;     sequences.
```

```
; UTIL.MCA
; Stereo IMPACT IDPU Flight Software
; Utilities Module
;
; Various utility routines:
;  WaitUS:  Wait W0 microseconds (processor cycle counter)
;  InitError: Initialize error handler
;  LogError:  Log an error, code in B0, data in B1.
;     - a count of errors, plus the most recent code
;       and data are maintainer & reported in housekeeping.
```

```
; EBOOT.MCA
; Stereo IMPACT IDPU Flight Software
; EEPROM Boot Module (called after PROM boot)
;
; This code initializes the modules not included in
; boot PROM, then starts up an expanded foreground
; polling loop that includes both PROM tasks and
; the non-PROM tasks.
```

```
; Mag.MCA
; Stereo IMPACT IDPU Flight Software
; Magnetometer instrument handler
; This module contains:
;   InitMag:  Initialize the MAG handler (on EBOOT),
;       including the MAG instrument data buffer where
;       MAG data is sent when received over the MAG
;       serial interface
;   PollMag:  Poll the mag system (from the foreground loop)
;       See if there is any data in the MAG telemetry buffer
;       and process it into Telemetry, burst, beacon,
;       and SWEA averages.
;       Also runs auto-ranging and updates MAG controls.
;   MagCommand:  Command handler for commands in the Mag
;     ApID range.  Called by command router (in foreground)
```

```
; SEP.MCA
; Stereo IMPACT IDPU Flight Software
; SEP Instrument suite interface handler
; Module includes:
;   InitSEP:  Initialize SEP system (if status indicates
;       it is powered up).  On boot and by command.
;   PollSEP:  called by foreground polling loop.  Check
;       to see if there is any new SEP data in the SEP buffer.
;       Parse and process (telemetry packets go to telemetry
;       queue, Housekeeping goes to housekeeping sample buffer,
;       Beacon goes to beacon sample buffer).
;   SEPCommand:  Command handler for commands in the SEP
;       ApID range.  Called by command router (in foreground)
;       Queue commands to be forwarded to SEP via InstCtrl
;       nodule (at 128Hz under TimerTic interrupt)
;   GetSEPCommandWord:  Get the next SEP command word
;       from the queue (filled by SEPCommand).  Called from
;       InstCtrl at 128Hz (by TimerTic interrupt)
;       Note that the timer tic is used to spread the SEP
;       commands out to avoid backing up the SEP processor.
;   SEPInterfaceCommand:  Command handler for commands in
;       the  SEP Interface ApID range.  Called by command router
;       (in foreground).
```

```
; Burst.MCA
; Stereo IMPACT IDPU Flight Software
; Burst memory handler
; The burst memory (which includes most of the processor-
;   addressable RAM) is used to snapshot high time resolution
;   data from a subset of the instruments (MAG, SWEA, STE).
; A trigger system determines when an interesting event has
;   occurred, and generates a metric for how interesting an
;   event it is.
; A set of three buffers are used:  One is being slowly
;   telemetred, one is contains the best burst seen so far,
;   (since the previous best burst started transmission),
;   and the third continuously recording while looking for
;   and even better burst.
; Burst data is saved continuously so that a burst precursor
;   of data taken before the event was triggered is collected.
;   In addition data is collected after the burst.
; The pre- and post-trigger buffers are fixed size, giving
;   fixed length (duration) bursts.  The sizes can be adjusted
;   by command within the constraints of the memory
; There are three precursor buffers and two post-buffers.  When
;   it is determined that a new burst has been triggered that is
;   even better than the one currently saved, the post
;   trigger buffer of the previous event is over-written
;   with the new event.
; The burst system hands out memory pointers to burst packet
;   generators.  Burst packets never cross memory page
;   boundaries.
```

```
; Burst data is saved as CCSDS telemetry packets in the burst
;   memory.
; Burst telemetry sources must format the packets into the
;   memory buffer provided by the burst system.
;
; This module contains:
;  InitBurst:  Initialize the burst system (on boot)
;  PollBurst:  Foreground polling loop entry.
;    PollBurstSend:  Decide if it is time to queue another telemetry
;      packet from the burst memory.
;    PollBurstTrigger:  See if a new burst has triggered
;  NewBurstPacket:  Called by a burst telemetry generator to
;    get a pointer to a new packet  pointer.
;  BurstCommand:  Command handler for commands in the
;    Burst ApID range.  Called by command router (in
;    foreground)
```

```
;  STE.MCA
;  Stereo IMPACT IDPU Flight Software
;  STE Instrument suite interface handler
;  Module includes:
;    InitSTE:  Initialize STE system (if status indicates
;       it is powered up).  On boot and by command.
;    PollSTE:  called by foreground polling loop.  Check
;       to see if there is any new STE data in the STE-U buffer.
;       Parse and process.
;       Also refresh STE-U and STE-D registers & LUT
;    ProcessSTEDMessage:  Process an STE-D message parsed from
;       the SWEA-STED buffer by the SWEA module
;    STECommand:  Command handler for commands in the STE
;       ApID range.  Called by command router (in foreground)
```

IMPACT

IDPU COMMAND AND TELEMETRY

File: IMPACT_CTM_B.xls

Revision: 2002-Aug-8

Sheet	Name	Sheet	Name		
1	APIDS	6	TABLES	11	ApID 201 (Dump)
2	TMSOH	7	Definitions	12	ApID 208,209 (MAG)
3	CMDS	8	ADC	13	ApID 220, 221 (STE Monitor Rate
4	CmdSeq	9	YSICONV	14	ApID 222,223 (STE Spectra)
5	ERRORS	10	Compress (19-8)	15	

IMPACT TELEMETRY APPLICATION IDENTIFIERS						
(Hex)	Description of Packet	Module				
200-207	**System**					
200	Housekeeping	HKP				
201	Memory Dump	System				
208-20F	**MAG**					
208	MAG Telemetry	MAG				
209	MAG Burst Telemetry	MAG				
210-21F	**SWEA**					
220-22F	**STE**					
220	STE Monitor Rates	STE				
221	STE Burst Monitor Rates	STE				
222	STE Spectra	STE				
223	STE Burst Spectra	STE				
230-23F	**Spare**					
240-26F	**SEP**					
270-27F	**IMPACT Beacon**					
300-36F	**PLASTIC**					
370-37F	**PLASTIC Beacon**					

IMPACT IDPU TELEMETRY FORMAT (SOH)												
IDPU SOH Start	ApID	200		Total SOH Length=		272	OK					
Addressing												
IDENTIFIER	**Lng**	**OFFSET**	**TYPE**	**WIDTH**	**A1**	**A0**	**Units**	**RL**	**YL**	**YH**	**RH**	**DESCRIPTION**
UTCs	4	6	U1234	32	1	0	Sec					UTC Seconds since epoch
UTCf	2	10	U12	16	1.53E-05	0	Sec					UTC Seconds fraction (2^-16 sec)
ISampleTime	2	12	TIMEHMS	16	1	0						Sample Time, Hours, Min, Sec (4,6,6 bits)
ISoftwareVersion	1	14	UB	8	1	0		-1	20	256	256	Flight Software Version Number
IHardwareVersion	1	15	UB	8	1	0						Hardware Version Number (FPGA)
IPollTime	2	16	U21	16	0.000032	0	Sec	-1	0	0.8	1	Maximum foreground loop time
IErrorCode	1	18	UB	8	1	0						Error Code
IErrorData	1	19	UB	8	1	0						Error Data
IErrorCount	1	20	UB	8	1	0		-1	-1	1	1	Error Count
ICommandCount	1	21	UB	8	1	0						Command Counter
ICommandLastID	1	22	UB	8	1	0						Last Command Ap ID (Bits 8,6,5,4,3,2,1,0)
ICommandLastSeq	1	23	UB	8	1	0						Last Command Sequence Counter (LSB)
IIMPACT_SSR	1	24	UB	8	1	0	%	-1	-1	99	100	IMPACT SSR Fill level
IPLASTIC_SSR	1	25	UB	8	1	0	%	-1	-1	99	100	PLASTIC SSR Fill Level
ISCStatus	1	26	UB	8	1	0						Spacecraft Status (see below)
IBootSelect	1	27	UB	8	1	0		-1	-1	4	256	Boot Select (0-3=EEPROM, else PROM)
ICodePage	1	28	UB	8	1	0		-1	0	64	64	Code Page (0=PROM, 8-11=EEPROM0-3, 16=RAM, Else = Illega
ICodePageStatus	1	29	UB	8	1	0						Code Page Status Bits (see below)
ICmdSeqRunning	1	30	UB	8	1	0		-1	-1	32	256	Command Sequence # Running (>31 -> no sequence running)
InterfaceEnables	1	31	UB	8	1	0						Instrument Interface Enables (see Below)
ITelemEnables	2	32	U21	16	1	0						Telemetry Enables (see below)
Spare	8	34										
IMagHKP	2	42	U21	16	1	0						MAG Hardware Housekeeping (see below)
ISTEUA	32	44	U21	16								STE-U analog housekeeping
ISWEAMCP	2	76	U21	16			V					SWEA MCP Voltage
ISWEANR	2	78	U21	16			V					SWEA Non-regulated HV Voltage
ISWEAAnal	2	80	U21	16			V					SWEA Analyzer Voltage
ISWEADefl1	2	82	U21	16			V					SWEA Defelector 1 Voltage
ISWEADefl2	2	84	U21	16			V					SWEA Deflector 2 Voltage
ISWEAV0	2	86	U21	16			V					SWEA V0 Voltage
ISWEAAnalCmd	2	88	U21	16			V					SWEA Analyzer Control Voltage
ISWEASTEDVCC	2	90	U21	16	-0.00508	9.908	V	4.5	4.75	5.25	5.5	SWEA/STE FPGA Interface +5V Supply voltage (ref)
ISWEATemp	2	92	U21	16	*		C					SWEA MCP Temperature
ISTEDTemp	2	94	U21	16	*		C					STE-D temperature
ISWEAPWBTemp	2	96	U21	16	*		C					SWEA PWB Temperature
ISWEASTEDVCCA	2	98	U21	16	0.001221	0.02	V	2.25	2.375	2.625	2.75	SWEA/STE-D +2.5V Supply Voltage

IMPACT IDPU COMMAND CODES

IDENTIFIER	ApID / Root	Value	Value	Value	Value	Value		Description
SYSTEM	0x200						;	System command root
MAG	0x210						;	MAG Command Root
SWEA	0x220						;	SWEA command root
STE	0x230						;	STE command root
Burst	0x240						;	Burst command root
Spare	0x250						;	spare
SEP	0x260						;	SEP command root
SEPInterface	0x270						;	SEPInterface
SYSTEMP	0x300						;	System command root (Only if enabled)
PLASTIC	0x310						;	PLASTIC command root
Spare	0x320						;	spare
SystemTableLoad	0x200						;	Load system table (offset8, value…)
SystemFunction	0x201						;	Execute system function (code8, value…)
IDPUMemoryLoad	0x202						;	Load memory (page8, offset16, value…)
IDPUEEPROMLoad	0x203						;	Load EEPROM (page8, offset16, value…)
DumpSystemTable	SystemFunction	1					;	Dump the System table
DumpMemory	SystemFunction	2					;	Dump memory (page8, offset16, count16) (Count = 0xFFFF -> repeat)
ExecuteImmediate	SystemFunction	3					;	Jump to address (address16)
ExecutePolled	SystemFunction	4					;	Poll an address (address16)
SetCodePage	SystemFunction	5					;	Set code page (page # 8: 0-3=EEPROM, 4=RAM) CAUTION
CopyEEPROMtoRAM	SystemFunction	6					;	Copy EEPROM code page to RAM code page (EEPROM page # 8, 0-3)
UpdateChecksum	SystemFunction	7					;	Update a code page checksum (page # 8: 0-3=EEPROM, 4=RAM)
EEPROMEnable	SystemFunction	8					;	Enable EEPROM Write
EEPROMDisable	SystemFunction	9					;	Disable EEPROM Write
ClearErrors	SystemFunction	10					;	Clear error counter
InstCtrl	SystemFunction	11					;	Send a control to an instrument: (Mask8, Reg8, Data16)
StartCmdSeq	SystemFunction	12					;	Start a command sequence (Sequence8)
DelayCmdSeq	SystemFunction	13					;	Delay command sequence (Seconds8)
EnablePLASTICSysCmd	SystemFunction	14					;	Enable PLASTIC System Commands (ApID 0x300-0x30F, mapped to 0x200 to 0x20F)
DisablePLASTICSysCmc	SystemFunction	15					;	Disable PLASTIC System Commands (ApID 0x300-0x30F, mapped to 0x200 to 0x20F)
ShutdownIDPU	SystemFunction	16					;	Request spacecraft turn off IDPU
ShutdownSEP	SystemFunction	17					;	Request spacecraft turn off SEP
ShutdownSWEA	SystemFunction	18					;	Request spacecraft turn off SWEA
ShutdownPLASTIC	SystemFunction	19					;	Request spacecraft turn off PLASTIC

IMPACT IDPU COMMAND SEQUENCES

32 command sequence tables, started by events or by command

Each sequence contains up to 128 bytes of commands

Commands are formatted like normal commands, except the CCSDS header is replaced with a 2-byte header:

- MSB = command size in bytes, excluding this header
- LSB = coded ApID: 0-7Fh -> 200-27Fh (IMPACT) , 80h-FFh -> 300-37FH (PLASTIC)
- Header = 0 -> end of sequence

Any number of sequences can be requested at one time (only one request per sequence number)

- Sequences run to completion. On completion, if multiple sequences are requested, the highest numbered sequence runs.

Table	Event	Module	Description
0	IDPUPwrWarn	Status	Spacecraft IDPU power-off warning
1	IDPUPwrWarnClr	Status	Spacecraft IDPU power off warning cleared
2	SEPPwrWarn	Status	Spacecraft SEP power-off warning
3	SEPPwrWarnClr	Status	Spacecraft SEP power off warning cleared
4	SWEAPwrWarn	Status	Spacecraft SWEA power-off warning
5	SWEAPwrWarnClr	Status	Spacecraft SWEA power off warning cleared
6	PLASTICPwrWarn	Status	Spacecraft PLASTIC power-off warning
7	PLASTICPwrWarnClr	Status	Spacecraft PLASTIC power off warning cleared
8	ThrusterWarn	Status	Spacecraft thruster warning
9	ThrusterWarnClr	Status	Spacecraft thruster warning cleared
10	FinePointingWarnClr	Status	Spacecraft fine pointing gone good
11	FinePointingWarnClr	Status	Spacecraft Fine pointing gone bad
12			
13			
14			
15			

IMPACT IDPU ERRORS		
8/20/2002		Note: append data to description.
Error Name	**Error Code**	**Description**
; Boot/1553:		
ERROR_BadEEPROMChecksum	1	;Boot: Check of EEPROM on boot failed. Data=0
ERROR_1553IdleTimeout	2	;DRV1553: On initialization, timed out waiting for Summit chip to go idle. Data=0
ERROR_1553BITTimeout	3	;DRV1553: On initialization, Summit BIT took too long. Data=1553 Status register LSB
ERROR_1553BIT	4	;DRV1553: On initialization, Summit BIT failed. Data=8MSB of BIT register
ERROR_1553YF	5	;DRV1553: Received an interrupt indicating a Summit chip YF condition. Data=8 MSB of 1553 INTPEND register (promote bit 7 to ni
ERROR_1553MSG	6	;DRV1553: Received a message error. Data = Message ID (3 LSB), Message Info Word LSB (5 MSB)
ERROR_1553Timeout	7	;Watchdog: No 1553 activity timeout. Data=IRQH_Active (Bit 0=timer, 1=1553, 2=1st timeout, 3=2nd timeout)
ERROR_1553DCBBad	8	;DRV1553: DCB check failed (causes re-init). Data=buffer ID
; Command system:		
ERROR_CmdFLTCOverrun	10	;COMMAND: FLTC queue over-run (from CmdRxHandler). Data=0
ERROR_CmdFLTCFmt	11	;COMMAND: FLTC bad format. Data=LSB of first FLTC word
ERROR_CmdSyncError	12	;COMMAND: FLTC sequence error. Data=FLTC sequence number
ERROR_CmdPacketHeader	13	;COMMAND: Command Packet Header format rejected. Data=FLTC sequence number
ERROR_CmdBadApID	14	;COMMAND: Command Packet ApID not for IMPACT or PLASTIC. Data=ApID LSB
ERROR_CmdBadLength	15	;SYSTEM: Bad command length (for selected function) Data = ApID LSB.
ERROR_CmdBadFunction	16	;SYSTEM: "Function" command with illegal function code. Data=function code
ERROR_CmdBadTableEntry	17	;SYSTEM: Table load requeted for entry past end of table (data = entry #)
ERROR_CmdLoadFailed	18	;SYSTEM: Memory or table load command failed (data read back not what was written). Data=Address LSB
ERROR_CmdEEPROMOverrun	19	;EEPROM: EEPROM Write request, but EEPROM queue full. Data=Address LSB
ERROR_CmdEEPROMBadPage	20	;EEPROM: EEPROM Write request, Memory page selected is not an EEPROM page. Data=page selected.
ERROR_CmdEEPROMBadAddr	21	;EEPROM: EEPROM Write request, Memory address > page size
ERROR_CmdEEPROMBadPtr	22	;EEPROM: EEPROM Write request: Write queue pointer illegal. Write queue is flushed.. Data=0
ERROR_CmdEEPROMVerify	23	;EEPROM: EEPROM Write request, read-back after write verify failed. Data=number of failures
ERROR_CmdEEPROMRunning	24	;EEPROM: EEPROM Write Request while running code from EEPROM
ERROR_CmdEEPROMBusy	25	;EEPROM: Attempt to update EEPROM checksum while there is still EEPROM data in the write queue. Data=0
ERROR_CmdBadPage	26	;EEPROM: ChangeCodePage command with illegal code page selected (>4). Data=page requested.
ERROR_CmdSEPDisabled	27	;SEP: Attempt to send command to SEP when it is disabled (off). Data=0
ERROR_CmdSEPOverrun	28	;SEP: SEP command queue over-run. Data=0
ERROR_CmdBadSeqReq	29	;CmdSeq: Bad command sequence number requested (>31). Data = sequence number requested.
ERROR_CmdPLASTICSysDis	30	;SYSTEM: PLASTIC System Commands disabled, data = command APID LSB
; Telemetry system:		
ERROR_IMPACTTlmQueue	31	;TELEM: IMPACT telemetry queue failure. More than 256 packets queued. Data=0.
ERROR_PLASTICTlmQueue	32	;TELEM: PLASTIC telemetry queue failure. More than 256 packets queued. Data=0.
ERROR_HSKPktNotReady	33	;HKP: Housekeeping packet queue overrun. Data=0
ERROR_DumpPktNotReady	34	;SYSTEM: Memory/Table dump packet queue overrun. Data=0.

IMPACT IDPU PARAMETER TABLES

Table	Length	Description
1		System
2		MAG
3		
4		STE
5		Burst
6		
7		

IMPACT IDPU PARAMETER TABLES

System Table (ID=1)

Offset	Length	Name	Conversion	Default	Description
0	1	BootSelect		0	0-3=EEPROM images, else PROM
1	1	HKPPacketRate		60	Housekeeping packet rate, seconds, 60=max.
2	1	DumpPacketRate		60	Memory Dump packet rate, seconds, 60=max
3	6	Spare			
9	1	Checksum			

MAG Table (ID=2) =System Table +0Ah

Offset	Length	Name	Conversion	Default	Description
0	2	MagRangeLower		6400	Autoranging lower threshold (Max<Lower -> high range)
2	2	MagRangeUpper		28656	Autoranging upper threshold (Max>Upper -> low range)
4	1	MagTlmAvg		4	Number of samples to average in telemetry packets (32Hz/4 = 8Hz
5	1	MagSensorMode		81h	Sensor mode. 7=Range, 6=IFC, 5=ADCCal, 0=Autorange
6	7	Spare			
13	1	Checksum			

IMPACT Memory Dump Packet							APID 201h
Module	Format	Length	Description				
SYSTEM	0 0 0 1 0 0 1 0	1	Version=000, Type=1, Secondary Header=1, Apid == 201				
SYSTEM	0 0 0 0 0 0 0 1	1					
SYSTEM	1 1 c c c c c c	1	Packet Sequence Control = 11B.				
SYSTEM	c c c c c c c c	1	APID Counter Field = 'c' counts each APID separately.				
SYSTEM	0 0 0 0 0 1 0 0	1	Application Data Field length (less 1) =	265	0109	hex	
SYSTEM	0 1 0 0 0 0 1 1	1					
SYSTEM	Clock MSByte	1	Secondary Header Format				
SYSTEM	Clock	1					
SYSTEM	Clock	1					
SYSTEM	Clock (LS Byte)	1					
SYSTEM	Subsecs (MSB)	1					
SYSTEM	Subsecs (LSB)	1	Extra byte of time				
SYSTEM	Dump Table	1	Dump Table ID (see below)				
SYSTEM	Dump Page	1	Dump page				
SYSTEM	Dump Adr	1	Dump Address, LSB				
SYSTEM	Dump Adr	1	Dump Address, MSB				
SYSTEM		256	Dump data				
SYSTEM							
SYSTEM							
SYSTEM							
SYSTEM							
SYSTEM							
SYSTEM							
SYSTEM							
		272	**Total Length of the Packet**				

IMPACT MAG Telemetry Packet						APID 208h, 209h
Module	Format	Length	Description			
SYSTEM	0 0 0 1 0 0 1 0	1	Version=000, Type=1, Secondary Header=1, Apid = 208h or 209h			
SYSTEM	0 0 0 0 0 0 0 1	1				
SYSTEM	1 1 c c c c c c	1	Packet Sequence Control = 11B.			
SYSTEM	c c c c c c c c	1	APID Counter Field = 'c' counts each APID separately.			
SYSTEM	0 0 0 0 0 1 0 0	1	Application Data Field length (less 1) =	265	0109	hex
SYSTEM	0 1 0 0 0 0 1 1	1				
SYSTEM	Clock MSByte	1	Secondary Header Format			
SYSTEM	Clock	1	Time stamp of first MAG sample averaged into first telemetry sample			
SYSTEM	Clock	1				
SYSTEM	Clock (LS Byte)	1				
SYSTEM	Subsecs (MSB)	1				
MAG	FirstSample	1	Sample number of sample second for first sample in packet (0-31)			
MAG	MagTlmAvg	1	Average count, from Mag table, 1,2,4,8,16, or 32 samples per average (1 for burst)			
MAG	MagStatus	1	MAG status from first sample (cal & range bits)			
MAG	X	2	First sample average, Mag X, MAG21 format			
MAG	Y	2	First sample average, Mag Y, MAG21 format			
MAG	Z	2	First sample average, Mag Z, MAG21 format			
MAG		252	42 more MAG samples averages, like the first one, in time order			
MAG						
MAg						
		272	**Total Length of the Packet**			

APID 208 = Normal telemetry
ApID 209 = Burst Telemetry

IMPACT STE Monitor Rates Telemetry Packet				APID 220h, 221h
Module	**Format**	**Length**	**Description**	
SYSTEM	0 0 0 1 0 0 1 0	1	Version=000, Type=1, Secondary Header=1, Apid = 220h or 221h	
SYSTEM	0 0 0 0 0 0 0 1	1		
SYSTEM	1 1 c c c c c c	1	Packet Sequence Control = 11B.	
SYSTEM	c c c c c c c c	1	APID Counter Field = 'c' counts each APID separately.	
SYSTEM	0 0 0 0 0 1 0 0	1	Application Data Field length (less 1) = 265 0109 hex	
SYSTEM	0 1 0 0 0 0 1 1	1		
SYSTEM	Clock MSByte	1	Secondary Header Format	
SYSTEM	Clock	1	Time stamp of the start of the accumulation interval for the first Monitor Rates sample	
SYSTEM	Clock	1		
SYSTEM	Clock (LS Byte)	1		
SYSTEM	Subsecs (MSB)	1		
STE	STERatesInterval	1	Accumulation time per sample, 2,10,30,60 seconds	
STE		20	Spare	
STE		1	STE-U Monitor Rates, Log compressed	
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE		1		
STE	STED Monitor rates	12	STE-D Monitor Rates, same as above for 4 STE-D detectors	
STE		216	9 more 24-byte monitor rates samples sets for consecutive STERatesInterval	
		272	**Total Length of the Packet**	

IMPACT STE Spectra Telemetry Packet				APID 222h, 223h
Module	**Format**	**Length**	**Description**	
SYSTEM	0 0 0 1 0 0 1 0	1	Version=000, Type=1, Secondary Header=1, Apid = 220h or 221h	
SYSTEM	0 0 0 0 0 0 0 1	1		
SYSTEM	1 1 c c c c c c	1	Packet Sequence Control = 11B.	
SYSTEM	c c c c c c c c	1	APID Counter Field = 'c' counts each APID separately.	
SYSTEM	0 0 0 0 0 1 0 0	1	Application Data Field length (less 1) = 265 0109 hex	
SYSTEM	0 1 0 0 0 0 1 1	1		
SYSTEM	Clock MSByte	1	Secondary Header Format	
SYSTEM	Clock	1	Time stamp of the start of the accumulation interval for the first Spectra sample	
SYSTEM	Clock	1		
SYSTEM	Clock (LS Byte)	1		
SYSTEM	Subsecs (MSB)	1		
STE	STESpecInterval	1	Accumulation time per sample, 2,10,30,60 seconds	
STE	STENumEnergy	1	Number of STE Energies (16,32,64)	
STE	Spare	1	Spare	
STE	Spare	1	Spare	
STE	STEStart	1	Number of 16-byte words remaining from sample in previous packet (only used in 64E mode)	
STE	Det0E0	1	Counts for Detector 0 Energy 0, Log compressed	
STE	Det0E1	1	Counts for Detector 0 Energy 1, Log compressed	
	…	29		
STE	Det0En	1	Counts for Detector 0 Energy STENumEnergy-1, Log compressed	
STE	Det1E0	1	Counts for Detector 0 Energy 0, Log compressed	
STE	Det1E1	1	Counts for Detector 0 Energy 1, Log compressed	
	…	29		
STE	Det1En	1	Counts for Detector 0 Energy STENumEnergy-1, Log compressed	
STE		192	6 more STENumEnergy-byte sample sets for consecutive detectors	
STE			Note: For STENumEnergies=32, one set of 8 detectors * 32 Energies per packet	
STE			For STENumEnergies = 16, two sets of 8 detectors * 16 energies per packet	
STE			For STENumEnergies = 4 detectors * 64 energies per packet	
			STEStart = 0 -> detectors 0-3, STEStart=128 -> detectors 4-7	
			Other values of STEStart indicate a loss of synchronization which may be recoverable.	
		272	**Total Length of the Packet**	