SETS Application (Version 1.0) Manual
Version 1.1 Update

V. Tolat

September 18, 1991

1 Introduction

The SETS application is the program which provides command and control, housekeeping, and telemetry for the SETS experiment. It is organized as a single monolithic program which spawns up to five processes that communicate through shared memory. In flight configuration, however, only four processes are created. The five processes (six including the parent process) perform different functions. The main (parent) process, after creating the other processes, accepts commands from its default input file (stdin). These commands are written to the virtual command device. The other processes perform the functions of housekeeping, time management, limit checking and command execution. The fifth process, which is not used during flight, saves telemetry into a file.

1.1 Master/Slave mode

The SETS application can operate in one of two primary modes : master and slave. In flight configuration, the SETS application runs in master mode on one cpu and in slave mode on the other cpu. In master mode the program accepts and executes all default commands as well as commands directed to the host cpu, it collects housekeeping data, sends telemetry to the SDIO, receives time from the GMT pac and performs limit checking. In slave mode, the program accepts all commands but only executes commands directed at the host cpu and receives time from the GMT pac. None of other functions stated above are performed.

1.2 Commanding

Commands are received through a virtual command device (*/dev/cmd*) provided by the operating system. Commands are written to the device by the operating system kernel, another cpu, a separate application program or the SETS application program itself. Commands are read only by the SETS application when it is executing. If the SETS application or some other application which reads the command device is not executing, the kernel executes commands from the command device. After the SETS application is started, commands can be issued by the user through a serial port or to the SDIO via the SFMDM. Alternatively, commands can be issued with the `rcmd` program.

Commands are read and executed by the command process. Most commands are executed immediately by the command process itself, however, other commands which are time consuming are executing by a child process at a lower priority. The child process terminates upon completion of the command. Such a command is the **send** command which copies a file to the burst mode channel. All application commands are executed by a separate process.

1.3 Housekeeping

The collection of housekeeping data is performed by the housekeeping process. When started, the housekeeping process initializes its data structures and gains access to the data collecting devices. The housekeeping process installs interrupt handlers to service the AD0/1 boards and the SDIO boards. Once a second, at the time of the first synch interrupt after the request for minor frame 23, the housekeeping process is awakened to collect data.

High-rate science data (24 samples per second) is collected by the synch interrupt handler. Under nominal conditions, the 24th synch interrupt in a second will occur shortly after the request for minor frame 23. If this happens, then the new housekeeping data collected during the last second is transferred to a buffer for transmission during the next second. If the minor frame requests are too slow, the value of *mfcnt*, which is the number of synch interrupts received between requests for minor frame 23, will be greater than 24. If the requests are too fast, the value of *mfcnt* will be 24, however, minor frames in adjacent telemetry pages will be repeated. In either case, the time fields in the telemetry page will not be incrementing properly.

1.4 Telemetry

Telemetry is sent to the SDIO board by an SDIO interrupt handler installed by the housekeeping process. The SDIO board is polled 48 times a second for minor frame requests. The SDIO board is polled after every synch interrupt and after interrupt by the AD0 board. Under nominal conditions, each of these interrupts occurs at 1/24 second intervals but are temporally shifted by 1/48 second. When a request is present, the minor frame number is read and that minor frame is written to the SDIO board. At this time, a command, if present, is read from the SDIO board and written to the command device.

1.5 Limit checking

Limit checking is performed by a limit checking process which is started by specifying a switch in the invocation of the SETS application. If the process is not started, no limit checking can be performed unless the application is killed and then restarted with the required switch. The limit process wakes up once a second and checks the limits of entries in the limit checking table. Only those entries for which limit checking is enabled are checked. Limit checking can be enabled/disabled for all entries, separately for each entry, or separately for the upper and lower limits of each entry.

1.6 Time management

Internal management of time is provided by a time management process. This process wakes up once a second. It first checks the last time the GMT was updated. If it has been more than a second, it updates the GMT based on the internal cpu clock. The time process then increments or decrements the time dependent variables. Under nominal conditions, i.e. when the GMT pac is operating and receiving time from the SFMDM, the variables are adjusted at a regular one second rate. If cpu time is being used to drive the GMT, the variables will be adjusted at irregular intervals (in a jumpy fashion) but by the correct amounts.

1.7 Command timeline

A command queue is available to timeline commands. The queue is managed by the command process. The granularity of time in the queue is one second. Commands are available to stop, start, hold, clear and list the timeline as well as to delete, hold, and release individual timeline entries.

1.8 File process

The file process, if started, stores telemetry in a file for later inspection. This process is awakened by the minor frame interrupt handler after every 24 synch interrupts. After storing the current telemetry page, the process sets the minor frame request to 23 in order to trigger the housekeeping process to begin sampling again after the next 24 synch interrupts. This process should not be started if the SFMDM is requesting minor frames via the SDIO since it modifies the minor frame request value.

1.9 Parameter (Logical) Table

A table of 100 logical parameters is provided. If an argument begins with a dollar sign (`$keyword`), the parameter table is searched for the keyword and a replacement is made if found. If the parameter is not found, no replacement is made and the command is aborted. A parameter name is limited to 8 characters in length and its value is limited to 24 characters in length. Commands are available to enter, delete, and modify parameter table entries.

1.10 Variables

Variables are special cases of values that need to be manipulated by the user. The variable command is used to list and modify these values. Special action may be taken when changing a variable's value.

2 Starting SETS

The usage for the SETS application is

```
sets [-dhltabf] [(filename)] { master | slave }

-d debug mode
-h housekeeping
-l limit checking
-t time management
-a adpac housekeeping
-b stdout to burst mode
-f telemetry to file (filename)
```

Either the `master` or `slave` keyword must be present. The switches are optional. In flight configuration, the SETS application is started automatically by the operating system. The application executes in master mode on cpu 1 (a) and in slave mode on cpu 2 (b). The switches used are `dhtlab`. For correct telemetry operations, the "-a" switch must be specified since this will result in the AD0 board generating interrupts at regular intervals which are used to poll the SDIO board for minor frame requests.

If the "-b" switch is specified, the standard output of all processes except for the limit checking process is sent to the burst mode channel with and id of 10 decimal. The standard output of the limit checking process is sent to the burst mode channel with an id of 11 decimal.

3 Commands

A grammar like syntax is used below to describe the commands. Arguments enclosed in square brackets [] are optional. Arguments enclosed in parenthesis () are to be substituted with an appropriate value. Curly braces are used as group delimiters. A vertical bar is used to denote a logical or. Unless otherwise stated, all numerical values are hex. All commands are free formated with spaces as delimiters between arguments. Commands are case sensitive.

3.1 Command prefix

A command prefix may be added to each command in order to direct the command to a specific cpu or to all cpus. If no prefix is included, the command is only executed by the application executing in master mode. The format of the prefix is

```
@(cpu) (command)
```

where (command) is any valid command. If (cpu) is 0, the command is executed by all cpus, regardless of their mode, i.e., master or slave. If (cpu) is nonzero (1 or 2), the command is only executed by the cpu whose id matches (cpu), regardless of its mode.

3.2 Memory and I/O commands

Display memory

```
r mem (addr) (count)
```

Display (count) bytes of memory starting at (addr).

Display port

```
r port {b | w} (port)
```

Display byte or word value of I/O port (port).

Modify memory

```
m {bit | byte | word | long} (addr) (value)
```

Write (value) to memory starting at (addr). Value can be of type `bit`, `byte`, `word`, `long`. If type is `bit`, the value is the binary representation of a byte, e.g., 01110110. Any character other than a 1 or 0 is considered to be a don't care and causes no change in that bit. This command generates no output.

Modify port

```
o {byte | word} (port) (value)
```

Write (value) to the word or byte I/O port (port). This command generates no output.

3.3 Direct commands

All direct commands have the letter `d` as their first argument. The second argument specifies an instrument or device to which the command is directed. The generic direct command is

```
d (device) (arguments)
```

A/D Multiplexer board (ADMUX)

```
d admux (which) [-g[(gain)]] [(ch1) ... [(ch7)]]
```

Display the value of one or more admux channels. A gain (0-2) can be specified. The default gain is 0. If a gain is not specified, the current gain is printed. Valid values of (which) are `a`, `b`, `c`, `d`. The range of channels is `0-F`. Up to seven channels may be specified in one command. Each command opens and closes the admux device so that the gain value is reset each time the command is issued.

D/A Channel

```
d dac (which) (value)
```

Write to a dac channel. Valid values of (which) are `a0`, `a1`, `b0`, `b1`, `c0`, `c1`, `d0`, `d1`. (value) is a 16 bit hex value. This command generates no output.

Parallel I/O board (PIOB)

```
d piob (which) r
d piob (which) w (mask) (value)
```

Display the value of a piob port or write to the port. Valid values of (which) are `00`, `01`, `02`, `04`, `05`, `06`, `08`, `09`, `0a`. (mask) is a byte mask which is anded with (value) before being written to the port. The write command generates no output.

AD0/1 boards (AD) There are many commands for AD0 and AD1 boards. Each command is listed below and described in turn. The (which) value for each command can be one of 00, 01, 10, 11, for PAC0 and PAC1 on the AD0 board and PAC0 and PAC1 on the AD1 board, respectively. The first digit designates the board and the second digit designates a pac on the board. The four pacs correspond to the signals 00-TCM, 01-CMM, 10-TVMDC, 11-TVMAC. Since PAC0 on each board is being used to collect high rate science (TCM and TVMDC), the direct ad commands should only be used for PAC1 on either board.

```
d ad (which) setup
```

Display setup parameters for the specified pac. These include the last GMT and CPU time that samples were taken . The sample command used is also displayed. The sample command can have the following values:

- 0x10 : Sample command with user defined parameters.
- 0x1n : Preset sample command n.
- 0x20 : Fast sample with user defined parameters.
- 0x2n : Preset fast sample n.
- 0x30 : Samples loaded with load command.
- 0x40 : Ram test.

```
d ad (which) setup -p (period)
```

Set the sampling period. (period) is in the range 14:FFFFFFFF for AD1 and 190:FFFFFFFF for AD0. The period is specified as single 32 bit number (8 hex digits and leading zeros are not required), however, period is actually the multiplication of the upper 16 bits by the lower 16 bits. The counter in the pac is implemented as two separate 16 bit counters. For example, to specify a period of 60A40 cycles, the period specified in the command would be 1060A4, i.e., 10 * 60A4.

```
d ad (which) setup -r (addr)
```

Set starting ram address. (addr) is in the range 0:7FFF.

`d ad (which) setup -t (trigger mode)`

Set trigger mode. Valid values are 0:6.

`d ad (which) setup -w (wait time)`

Set the time to wait before beginning to sample. Valid values are in the range `A:FFFFFFFF`. The wait time is implemented with dual counters as is the period and the value of (wait time) should be the values that go into the counters.

`d ad (which) setup -s (samples)`

Set the number of samples to take. Valid values are in the range `0:8000`.

`d ad (which) setup -i {(expid)|+}`

Set the experiment id for the pac. Valid ids are in the range `0:FFFF`. If a "+" is substituted for the id, the current id is incremented.

`d ad (which) trigger`

Trigger the pac to begin sampling.

`d ad (which) sample (cmd)`

Write a sample command to the pac. If (cmd) is zero, the user supplied parameters are used. If (cmd) is non-zero, the specified preset is used. No output is generated.

`d ad (which) fast (cmd)`

Write a fast sample command to the pac. (cmd) has the same meaning as above. No output is generated.

`d ad (which) diag`

Display all register values.

`d ad (which) reg (reg num) [(value)]`

Display the value of register (reg num). If (value) is specified, (value) is written to the register. (reg num) and (value) are hex.

```
d ad (which) reset {fifo|adc}
```

Write a reset fifo or reset adc command to the pac.

```
d ad (which) adccal {b|i} (mode)
```

Write an adc calibrate command to the pac. `b` is for burst calibration. `i` is for interleaved calibration. (mode) is 0/1.

```
d ad (which) abort
```

Abort the current pac operation.

```
d ad (which) ready
```

Set the pac ready flag.

```
d ad (which) ramtest
```

Command the pac to perform a ram test.

```
d ad (which) file {(start)|-} {+(len)|(end)|-} (filename)
```

Store memory from the pac starting at the address (start) and ending at the address (end) or the address (start+len) to the file (filename). If the "-" character is used in the command, the start address and number of bytes to read is obtained from the pac's registers. The "-" character would be used in the post-trigger mode since the start address and amount of data taken is not known a priori.

```
d ad (which) send (start){+(len)|(end)} (id)
```

Send memory in the burst mode channel with the specified id. The arguments are the same as for `file` except that (filename) is replaced by (id). A process is forked to perform the send since it may take a substantial amount of time.

```
d ad (which) load (start) (filename)
```

Load pac memory starting at the address (start) from the file (filename).

```
d ad (which) hskp {0|1}
```

Set the pac housekeeping flag to the specified value. A 0 disables housekeeping and a 1 enables housekeeping. Housekeeping refers to the function of the pac collecting high-rate science and is performed by PAC0 on the AD0 and AD1 boards

CCP

```
d ccp {qdac | cdac} (value)
d ccp (relay)
```

The first command writes (value) to the specified D/A channel. The second command sets the relay (relay). The relay names are `qrst`, `ccal`, `cin`, `cout`, `cen`, `cdis`, `qen`, `qdis`, `qhi`, `qlo`, `chi`, `clo`. Each time a ccp command is executed, all relays are first reset. If (relay) is `off`, all relays are just reset.

FPEG

```
d fpeg (var) (value)
```

Set the value of (var) to (value) in the FPEG command buffer. This does not send a serial command to the FPEG.

```
d fpeg ready
```

Load the FPEG with the contents of the command buffer and leave the FPEG word enable bit high. The FPEG does not latch and execute the command until the word enable bit is set low.

```
d fpeg go
```

Execute the command that has been previously loaded into the FPEG by setting the word enable bit low. This must follow the `d fpeg ready` command above.

```
d fpeg send
```

This performs the functions of both the `d fpeg ready` and `d fpeg go` commands above.

```
d fpeg stop
```

This causes the FPEG to abort the current command and stop its activity. Zero is written into the command buffer for the variables `ontime`, `offtime`, `npulse` and the `stop` bit is set. The command is then sent to the FPEG and executed. If a command has been loaded and not executed, it is first executed.

A list of variables and their possible values is given below. The following vars are all binary, 0/1:

`encom`, `enontm`, `enoftm`, `enpul`, `run`, `runbar`, `gun1`, `gun2`, `fil`, `hv`

The following vars have more than two values:

- `gun - 0,1,2,3`
- `ontime - 0:1F`
- `oftime - 0:1F`
- `npulse - 0:F`

SRPA

```
d srpa acl (value) swp (value)
```

If (value) is greater than zero, the corresponding bits in the SRPA command byte are set.

TCVM

```
d tcvm (relay) {0|1}
d tcvm (var) (value)
```

The first command is used to turn a relay on (1) or off (0). The relay names are:

> `acg1`, `acg2`, `acg3`, `acg4`, `aclp1`, `aclp2`, `cmmcali`, `cmmcaln`, `cmmfc`, `cmmgl`, `cmmg2`, `cmmlp1`, `cmmlp2`, `hsk0`, `hsk1`, `hsk2`, `hsk3`, `mms`, `cms`, `prot`, `r2500k`, `r250k`, `r25k`, `shunt`, `tcmcal`, `tcmlp1`, `tcmlp2`, `tvmcalac`, `tvmcali`, `tvmcaln`, `tvmen`, `tvmfc`, `tvmg1`, `tvmg2`

The second command sets groups of relays depending on (var) and (value). The variable names and their values are:

- `acgain - 1, 10, 100, 1000`
- `aclpf - 12, 100, 1000`
- `cmmgain - 1, 10, 100`
- `cmmlpf - 10, 12, 1000`
- `tcmlpf - 10, 12, 1000`
- `dcgain - 1, 10, 100`

3.4 INT1 command

```
int1 (adnum) (tcvmrelay) (bit) (state)
```

This command is used to set/reset a tcvm relay upon receipt of a synch interrupt and trigger an AD1 pac to begin sampling. (adnum) should be either 01 or 11. (tcvmrelay) is the relay code of the desired relay to set/reset. (bit) is the bit number of this relay in the tcvm status word. (state) is either 0 or 1 depending on the new state of the relay. If the command is successful, (bit) will be set according to (state) in the tcvm status word. Before the command is issued however, the AD pac must be initialized to begin sampling on a trigger by write. The tcvmrelay names, codes and bit numbers are given in the table below. The tcvm housekeeping relays should not be set/reset with this command.

3.5 Sequence commands

```
s (filename) [ (arg1) (arg2) (arg3) (arg4) ]
```

Execute a sequence from the file (filename). Each command can be upto 62 characters in length. This is longer than the length of commands that are allowed to be sent via the SFMDM (30 characters). Optional arguments may be specified after the sequence file name. These arguments are stored in the parameter table as $1, $2, etc. The sequence file name is stored as parameter $0. These parameters are global to the SETS application and not local to a sequence so that when another sequence is called, the argument parameters are replaced. Care must be taken to not submit a sequence command through the command channel before an initiated sequence has completed or has been processed to a point where the argument parameters are no longer needed.

```
s abort
```

Abort a sequence. All currently executing sequences will be aborted.

```
w (seconds) [(milliseconds)]
```

This command causes the processing of the sequence to be delayed by (seconds) seconds and (millisecond) milliseconds. The milliseconds field is optional. This command is only valid in a sequence and is interpreted as an application command if issued outside of a sequence.

name	off	on	bit
acg1	44	45	0
acg2	42	43	1
acg3	40	41	2
acg4	3e	3f	3
aclp1	f	e	4
aclp2	d	c	5
cmmcali	11	10	6
cmmcaln	13	12	7
cmmfc	33	32	8
cmmg1	47	46	9
cmmg2	49	48	a
cmmlp1	17	16	b
cmmlp2	15	14	c
cms	1a	1b	d
mms	1c	1d	e
prot	4d	4c	f
r2500k	1e	1f	10
r250k	20	21	11
r25k	22	23	12
shunt	24	25	13
tcmcal	1	0	14
tcmlp1	5	4	15
tcmlp2	7	6	16
tvmcalac	38	39	17
tvmcali	9	8	18
tvmcaln	b	a	19
tvmen	4e	4f	1a
tvmfc	19	18	1b
tvmg1	3b	3a	1c
tvmg2	3d	3c	1d

Table 1: TCVM relays.

3.6 Timeline commands

```
t list
```

List the timeline. The current time is displayed, GMT and MET. Each command in the timeline is displayed along with its id and the GMT at which it will be executed. The number of free timeline entries is also displayed. If the -b switch is used to invoke the application (the standard output is written to the burst channel), the output of this command is written to the burst channel with an id of 12.

```
t (command) (ttype) (time)
t sub (cnum):1 (command)
t sub (cnum):2  (command)
t sub (cnum):0  (ttype) (time)
```

(ttype) is one of d, m, g. If (ttype) is m or g, then (time) has the format day/hr:min:secs. If (ttype) is d, then (time) a delta time in decimal seconds. The first command above allows the user to submit (command) to the timeline. The other commands allow the user to construct a command in a buffer before submitting it. (cnum) refers to one of eight available buffers, 0:7. If the next number is 1, the command buffer is cleared and (command) is written into the command buffer. If the number is 2, (command) is appended to the current command buffer. In either case, the current contents of the command buffer are displayed. If the number is 0, the command in the buffer is submitted to the timeline.

```
t stop
t restart
t abort
```

The above commands stop, restart, and abort the timeline, respectively.

```
t del (id)
t hold (id)
t release (id)
```

The above commands will delete, hold and release an entry in the timeline, respectively. (id) is the number of the entry to be operated on.

3.7 Variable commands

```
v
v (varname) (value)
```

The first command will display the variables and their values. Values preceded with a "0x" are hex while the others are decimal. The second command sets the variable (varname) to (value). (value) should be the same radix as displayed with a v command. The variables, their formats and the effects of changing them are listed below. If the -b switch is used to invoke the application (the standard output is written to the burst channel), the output of this command is written to the burst channel with an id of 14.

- **flt1** (hex) - flt1 value. Default value is 0.
- **flt2** (hex) - flt2 value. Default value is 0.
- **debug** (hex) - Determines what debug messages are printed. Default value is 1 if "-d" switch is specified in the invocation of the application, otherwise it is 0.
- **echo** (hex) - If not zero, commands from the command channel are echoed. If the second bit is set (bit 1), commands from sequences are echoed. Default value is 1. A value of 2 or 3 will cause all commands to be echoed.
- **omi, omittg, oci, ocittg, soci, socittg, stp, stpttg** (decimal) - Orbit and operation mode variables. Default values are 0.
- **gmtmet** (day/hrs:min:sec) - GMT of MET 0. Default value is 0/0:0:0.
- **mastergmt** (0/1) - Determines which gmt pac is used for time. The master sets the gmt pac registers appropriately. This variable should be changed for both the master and slave simultaneously by using the @0 command prefix. Default value is 0.
- **mastersdio** (0/1) - Determines the master sdio pac. The master sets the sdio pac's registers. This command should be issued to both cpus simultaneously as well. Default value is 0.

- **sdiomode** (hex) - Hex value of the sdio mode register. Only the master sets the sdio pac's registers. This variable is only used for debugging purposes. Default value is 7.

- **slaveflag** (0/1). If 0, the application is in master mode, otherwise it is in slave mode. The variable cannot be set with the "v" command; an "**i master**" command must be used. Default value is set at invocation.

- **tcvmhskp** (0/1) - Disable/enable the housekeeping process from polling the tcvm for data. Default value is 0.

- **filen, hven** (0/1) - Disable/enable the turning on of the FPEG filament or high voltage relay. Default values are 0.

- **cmsen, shunten, proten** (0/1) - Disable/enable the turning on of the cms, shunt, and prot relays, respectively in the TCVM. Default values are 0. These enables are ignored by the `int1` command.

- **tvmdcmode** (hex) - Determines the format of the TVMDC data. The upper four bits determine the type of format and the lower four bits determines the resolution of the data.

  - `0X` - Shift the 16 bit values right `X` times and send the lower 8 bits of the shifted values.

  - `1X` - Shift the 16 bit values right `X` times and send the difference between the last value and the current value.

  - `2X` - Send all 16 bits of the first 8 values. The lower 4 bits are ignored.

- **int1pd** (24 bit hex) - The period of INT1 interrupts generated by the SDIO.

- **tmfilettg** (decimal) - The number of seconds remaining for a fixed file to be sent in place of telemetry.

3.8 Limit checking commands

```
lim
lim {on | off}
```

```
lim (var)  [ -l (lower) ] [ -u (upper) ]
           [ -cl {0|1} ] [ -cu {0|1} ]
           [ -f (fault) ] [ -x (cmd) ]
```

The first command lists the entries in the limit table, an indication as to whether limit checking is enabled or disabled, and the pid of the limit checking process. If the -b switch is used to invoke the application (the standard output is written to the burst channel), the output of this command is written to the burst channel with an id of 13. The second command enables/disables limit checking for all entries in the limit table. The third commands sets the parameters for the individual entries in the limit table. The "-l" and "-u" switches are used to set the lower and upper limits, respectively. The "-cl" and "-cu" switches are used to enable/disable the checking of lower and upper limits, respectively. The "-f" switch is used to set the fault code. The "-x" switch is used to set the command which is executed if a limit is exceeded. The "-x" switch must be the last switch on in a command since all words after it are taken to be part of (cmd).

The following table lists the entries in the limit table and their default parameters. Limit checking is initially disabled for all entries.

The limits for **sdio, gmt, fpegcl, ega** are checked by special procedures. The upper and lower limits in the table are not used, however, checking for at least one of these limits (lower or upper) must be enabled for these entries to be limit checked at all.

3.9 Parameter commands

```
p
```

List the parameter table. If the -b switch is used to start the application (standard output is directed to /dev/bm), the output of this command is written to the burst channel with an id of 15.

```
p (param)
```

List the definition for the parameter (param).

```
p -a (param) (value)
```

name	lower	upper	fault	command
fpegcl	0	0	1	
iamag	0	0xff	2	
icccp	0	0xff	3	
idep	0	0xff	4	
ifpeg	0	0xff	5	
ihtr	0	0xff	6	
ispib	0	0xff	7	
isrpa	0	0xff	8	
itcvm	0	0xff	9	
qdu	0	0xff	a	
qdl	0	0xff	a	
tcm	0	0xff	b	
tcvmpa	0	0xff	c	
tcvmpb	0	0xff	d	
tcvmps	0	0xff	e	
tcvmtrla	0	0xff	f	
tcvmtrlb	0	0xff	10	
tdep1	0	0xff	12	
tdep2	0	0xff	13	
tggh1	0	0xff	14	
tggh2	0	0xff	15	
tgel	0	0xff	16	
filontime	0	255	17	
ccpontime	0	60	18	
sdio	0	0	19	
gmt	0	0	1a	
ega	0	0	1b	

Table 2: Limit table entries

Add (param) to the parameter table with assignment (value) if (param) is not in the table already.

```
p -d (param)
```

Delete (param) from the parameter table.

```
p (param) (value)
```

Assign (value) to (param) in the parameter table. If (param) is in the table, change its current value to (value). If it is not in the table, add (param) to the table with (value) as its definition.

```
p -l (filename)
```

Load the parameter table from (filename). The format of the file is

```
(param) (value)
```

with each pair being separated by newlines. The command

```
p -l /dev/null
```

can be used to clear the parameter table.

```
p -f (filename)
```

Store the parameter table to file named (filename). The format of the file is the same as above.

3.10 Initialization commands

```
i cpu (cpuid) (passwd)
```

Reboot the cpu. (cpuid) must match the id of the cpu executing the command. By default (passwd) is `gandalf`.

```
i master {0|1} (passwd)
```

Change the primary mode of the application. The value of `slaveflag` is changed to the compliment of the value specified. It can be displayed with the "v" command. This command should normally be used with a command prefix so that it is executed by a single cpu. There should never be more than one master. To interchange the role of two cpus, the master must first be directed to become a slave and then a slave can be directed to become a master. For example, to change cpu 1 from a master to a slave and cpu 2 from a slave to a master, send the following sequence of commands :

```
i master 0 (passwd)
@2 i master 1 (passwd)
```

3.11 File loading

```
load [-n] (filename) (size) [hex] [(eofword)]
```

This command is used to load a filename named (filename). If (filename) doesn't exist, a file of (size) maximum bytes (in hex) is created. (filename) is loaded with commands read from the command device. Commands read from the command device are inserted into the file until a command is read where the first character has an ascii value less than a space (20 hex). If an "eofword" is specified, the loading is terminated when the value of "eofword" is encountered as a command. For example, if the command

```
load junk 200 EOF
```

is submitted, the load will terminate when the string `EOF` is encountered. A character less than a space will still terminate the load. During the file loading period, the value of *msw* is set to 2. During this period, commands are not processed from the timeline. If "-n" is not specified, newlines are implicitly inserted after every command. If it is specified, no newlines are inserted and the text from commands is inserted into the file without change or additions. If "hex" is specified, the commands are interpreted as hex characters, two per byte. Valid pairs of hex characters are converted into bytes and inserted into the file. Newlines are not added if "hex" is specified regardless of whether the "-n" switch is present.

3.12 Telemetry Files

The contents of a file can be substituted for telemetry. This is done with the command

```
tmpage (filename) [(seconds)]
```

This command causes the contents of the file (filename) to be sent as telemetry for (seconds) seconds. If the number of seconds is not specified, telemetry from the file is sent until the entire contents of the file have been sent. The variable **tmfilettg** is set to the number of seconds remaining for substitution of telemetry. After one **tmpage** command, the same file can be sent as telemetry again by setting the **tmfilettg** variable to a non-zero value. To terminate file substitution, the **tmfilettg** variable should be set to 0.

3.13 Miscellaneous commands

Send to burst mode

```
send (filename) (id)
```

Send the contents of (filename) in the burst mode channel with an id of (id). The value of (id) should be greater than 20 hex.

Change directory

```
cd (dir)
```

Change the default directory to (dir).

CPU status

```
systat [(filename)]
```

Display the current cpu time, GMT and MET along with the process status and memory status. If (filename) is specified, the output is written to (filename). This is useful for getting the status of the slave cpu(s). To get the status of cpu 1, issue the following commands:

```
@1 systat /ram0/xx
send /ram0/xx (id)
```

Process status

```
ps
```

Display process status.

Memory status

```
memory
```

Display memory status.

Time

```
time
```

Display the current cpu time, GMT and MET.

3.14 Application commands

Any command which is not one of the above, is assumed to be the name of an application and an attempt is made to execute it as such. Some useful applications are listed below.

- `cat f1 .. fn` - list contents of one or more files
- `df` - display filesystem usage
- `hello` - print "Hello world"
- `count (xx)` - count up from 0 to (xx)
- `ls (dir)` - list directory (dir)
- `echo (args)` - echo argument list
- `eefile -a (file) (source)` - add file to eeprom; (file) is the eeprom filename and (source) is the file to copy; if (file) exists, it is first deleted; if (source) is a directory, a recursive copy is performed
- `eefile -d (file)` - delete a file from eeprom; if (file) is a directory, a recursive delete is performed

- `eefile -n (file) (newfile)` - rename an eeprom file (file) to (newfile); files cannot be renamed to a new directory; (newfile) specifies the filename only and not the path
- `rcmd (cmd device) (command)` - write command to (cmd device); (cmd device) can be either `/dev/cpua`, `/dev/cpub` or `/dev/cmd`
- `od [-cdox] (file)` - dump file in format specified (o:octal, d:decimal, x:hex, c:ascii)
- `mv f1 f2; mv f1 .. fn d1` - rename files
- `cp f1 f2; cp [-r] f1 .. fn d1` - copy files
- `rm [-r] f1 .. fn` - remove files
- `bcmp (addr1) (addr2) (cnt)` - compare (cnt) bytes from (addr1) and (addr2) and report the first set of bytes that are different
- `chksum (addr1) (addr2); chksum -f (file)` - report checksum for the address range or file
- `init (file)` - Initialize the system using the init file specified. To execute `init` from the SETS application, it must be, the following command must be used:

  ```
  shell q -t5 init (file)
  ```

  This allows `init` to be run as a process without a parent process. It is necessary for the proper operation of `init`.

4 Telemetry Data Format

The format of the telemetry page is shown in tables 3, 4 and 5.

Each of the telemetry fields are briefly described below. They are ordered and grouped according to functionality and not position in the telemetry page. Next to each variable name is its position in the telemetry page. The position is given as (x,y) where x is the minor frame (beginning with 0) and y is the column (beginning with 1).

Annex 4
Table 07.

Format of TFL192.

page 1-008

4 digit fields → gives Dcom data. start field, # of fields

Stripping done by PDI.
Orbiter stream data.
120 kbit
42 words / s frame
minor frame.

1 —3

5 -13

1	2	3	4	5	6	7	8	9	10	11	12
C0	3F	0	CP	QP	IPHI	IPLO	LP	GMT1	TCVMTRLC	TCVMM15	PWRSW
C0	3F	1	-	-	-	-	-	GMT2	TDEP1	TCVMP15	FPEGSW3
C0	3F	2	-	-	-	-	-	GMT3	GDQ1	TCVM5	MSW
C0	3F	3	-	-	-	-	-	GMT4	GDQ2	GP15	FLT1
C0	3F	4	-	-	-	-	-	MBUS1	GFOC1	G5	FLT2
C0	3F	5	-	-	-	-	-	MBUS2	GFOC2	GM15	OMI
C0	3F	6	-	-	-	-	-	MBUS3	GUNSYNCSW	G120	OMITTG1
C0	3F	7	-	-	-	-	-	MBUS4	TDEP2	G220	ISPIB
C0	3F	8	-	-	-	-	-	MBUS5	IAMAG	CCPP15	OCI
C0	3F	9	-	-	-	-	-	MBUS6	ICCP	CCPM15	OCITTG1
C0	3F	10	-	-	-	-	-	CPUTIME1	IHTR	CV	TGGH1
C0	3F	11	-	-	-	-	-	CPUTIME2	ISRPA	QV	SOCI
C0	3F	12	-	-	-	-	-	CPUTIME3	TCVMTEL	IPP6	SOCITTG1
C0	3F	13	-	-	-	-	-	CPUTIME4	TCVMTPS	IPM6	GHV1
C0	3F	14	-	-	-	-	-	FLT1CNT	TCVMTBP	IP10	STP
C0	3F	15	-	-	-	-	-	FPEGCNT	IPSWP4	FLT2CNT	STPTTG1
C0	3F	16	-	-	-	-	-	TSPIB2	TCVMTCAP	IP85	TCVMSW2
C0	3F	17	-	-	-	-	-	TSPIB3	TCB	IPSWP1	FPEGONTM
C0	3F	18	-	-	-	-	-	TCVMCMD	TCP	LPAC	FPEGOFFTM
C0	3F	19	-	-	-	-	-	FPEGSW1	TQE	LPSWP	FPEGNPULS
C0	3F	20	-	-	-	-	-	FPEGSW2	TIPA	VSPIB	CCPSW
C0	3F	21	-	-	-	-	-	SRPASW	TIPP	SBX	BX
C0	3F	22	-	-	-	-	-	MFCNT	TSPIB1	SBY	BY
C0	3F	23	-	-	-	-	-	IPSWP2	IPSWP3	SBZ	BZ

Table 3: Telemetry Page (1 of 3)

15 17 - 20

13	14	15	16	17	18	19	20	21 - 34
GAND1	CD	QD	QDCDLSN	TCM		TVMDC		TVMDC LSB
GAND2	-	-	-	MSB	LSB	MSB	LSB	-
GCAT1	-	-	-	-	-	-	-	-
GCAT2	-	-	-	-	-	-	-	-
IDEP	-	-	-	-	-	-	-	-
IFPEG	-	-	-	-	-	-	-	-
OMITTG2	-	-	-	-	-	-	-	-
ITCVM	-	-	-	-	-	-	-	-
TCVMTRLA	-	-	-	-	-	-	-	-
OCITTG2	-	-	-	-	-	-	-	-
TGGH2	-	-	-	-	-	-	-	-
TGEL	-	-	-	-	-	-	-	-
SOCITTG2	-	-	-	-	-	-	-	-
GHV2	-	-	-	-	-	-	-	-
TCVMSW1	-	-	-	-	-	-	-	-
STPTTG2	-	-	-	-	-	-	-	-
TCVMSW3	-	-	-	-	-	-	-	-
TCVMSW4	-	-	-	-	-	-	-	-
TCVMSW5	-	-	-	-	-	-	-	-
TCVMTRLB	-	-	-	-	-	-	-	-
TCVMPA	-	-	-	-	-	-	-	-
TCVMPB	-	-	-	-	-	-	-	-
ENABLES	-	-	-	-	-	-	-	-
TCVMP28	-	-	-	-	-	-	-	-

Table 4: Telemetry Page (2 of 3)

35 - 36	37 - 38	39 - 40	41 - 46	47 - 50	51	52	53	54	55 - 56
BM ID	BM SEQ	BM CNT							
BM DATA (512 bytes)									
				INT1PD	-	TVMDCMODE	FPEGONTIME	CCPONTIME	CHKSUM

Table 5: Telemetry Page (3 of 3)

GMT1 (0,9) Most significant byte of GMT in seconds.

GMT2 (1,9)

GMT3 (2,9)

GMT4 (3,9) Least significant byte of GMT.

MBUS1 (4,9) Most significant byte of the 48-bit multibus time. See section 4.1 below for more details.

It is tenths of milliseconds

MBUS2 (5,9)

MBUS3 (6,9)

MBUS4 (7,9)

MBUS5 (8,9)

MBUS6 (9,9) Least significant byte of multibus time.

CPUTIME1 (10,9) Most significant byte of the cpu time. See section 4.1 for more details.

CPUTIME2 (11,9)

CPUTIME3 (12,9)

CPUTIME4 (13,9) Least significant byte of the cpu time.

TCVMSW1 (14,13) First byte of software TCVM relay states. Each bit is described in order of increasing bit position (right to left).

- ACG1 - If ACG1=1, the ac gain is x1. This bit overrides ACG2, ACG3, and ACG4. It is set by a latching mechanical relay which shorts the negative feedback of an op-amp.
- ACG2 - If ACG2=1, and all other ACG bits are zero, the gain is x10. ACG2 indicates the state of a mechanical latching relay which connects a resistor from the input of the ac gain op-amp to signal return.
- ACG3 - If ACG3=1, ACG1=0, and ACG4=0, the ac gain is x100. If ACG2 is not zero, there is a 10 per cent increase in the gain. Set by a mechanical latching relay which connects a resistor from signal return to the opamp inverting input.
- ACG4 - If ACG4=1 and ACG1=0, the ac gain is x1000. If ACG2 or ACG3 are not zero, a small error in the gain occurs. Set by a mechanical latching relay which connects a resistor from the signal return to the inverting input of the ac gain opamp.
- ACLP1 - Set by a solid state relay which is software latched.
- ACLP2 - Set by a solid state relay which is software latched.
- CMMCALI - Indicates 1 when the inverting input of CMM is enabled for calibration input. The inverting input is the signal side which corresponds to the attenuator output. The calibrate signal CALA is supplied by a 12 bit DAC in the DEP.
- CMMCALN - Indicates 1 when the noninverting input of the CMM instrumentation amp is enabled for calibration. This input

corresponds to the signal return (ground) from the CMM attenuator and provides a means of testing for common mode rejection and for offsetting the CMM input. The calibration input CALB is provided by a separate 12 bit DAC in the DEP from the CALA DAC.

TCVMSW2 (16,12) Second byte of software TCVM relay states.

- CMMFC - The CMM Frequency Control is a 300 kHz low pass filter which removes the peaking at x1 gain in the instrumentation amplifier of CMM.
- CMMG1 - Indicates the state of a mechanical latching relay which sets the gain of the CMM preamp.
- CMMG2 - Indicates the state of a mechanical latching relay which sets the gain of the CMM preamp.
- CMMLP1 - Indicates the state of a solid state relay which sets the CMM low pass filter.
- CMMLP2 - Indicates the state of a solid state relay which sets the CMM low pass filter.
- CMS - Indicates when the TCVM controlled side of the CMS high voltage relay driver AND gate is enabled. A second input from the CMSDOL must also be present to close the relay. CMS is a high voltage, high vacuum momentary relay which enables the Current Mode Switch circuit. The CMS bit will not be set unless the CMSEN enable bit is first set by ground manual command. Both CMS and CMSDOL bits must be set true (i.e. =1), before the relay is actually closed.
- MMS - The Mechanical Mode Switch is a high voltage, high vacuum momentary relay which is in series with and follows the CMS relay.
- PROT - The PROT is a high voltage, high vacuum relay which connects a 200 nF, high voltage capacitor between the CMS and MMS relays. The operation of the PROT relay is similar to the CMS relay except there is no DOL control. A software enable, PROTEN, is required before a command will be accepted to close PROT

TCVMSW3 (16,13) Third byte of software TCVM relay states.

- R2500K - A software derived bit which indicates the state of the high voltage, high vacuum momentary relay which connects a 2500 kohm resistor bank between the MMS relay and the TCVM frame. Both CMS and MMS must be closed to flow current through this resistor. The R2500K relay is on the ground side of the resistor.
- R50K - Same as the R2500K except 250 kohms. Temperature is monitored by TCVMTRLC.
- R25K - Parallel load resistor bank with R2500K and R250K. Comprised by two 50kohm resistors in parallel. One resistor is monitored by TCVMTRLA and the other by TCVMTRLB.
- SHUNT - The SHUNT status bit indicates the TCVM input to the SHUNT relay driver AND gate is true. The SHUNTDOL bit must also be true before the SHUNT will close. A software enable bit, SHUNTEN, must be true before the SHUNT bit command will execute. The SHUNT connects a 15 ohm resistor in parallel with the other load resistors between MMS and the TCVM frame. The SHUNT relay is on the ground side of the 15 ohm resistor.
- TCMCAL - The Tether Current Monitor input can be switched to CALB through a momentary mechanical relay. CALB is a DAC in the DEP and is the same DAC used for CMMCALN and TVMCALN.
- TCMLP1 - Software derived indicator of the state of a solid state relay which sets a low pass filter.
- TCMLP2 - Same as TCMLP1.
- TVMCALAC - The signal side input to the TVMDC preamp (inverting input) can be connected to one of two calibrate signals: CALA or CALC. The TVMCALAC bit indicates 1 for CALA and 0 for CALC, but TVMCALI must be 1 for calibration to be enabled.

TCVMSW4 (17,13) Fourth byte of software TCVM relay states.

- TVMCALI - Indicates the state of a momentary mechanical relay connected to the inverting (signal side) input of the TVMDC preamp. This is the input directly connected to the TVM attenuator.

- TVMCALN - Indicates the state of a momentary mechanical relay connected to the noninverting (reference side) input of the TVMDC preamp. This is the input connected to the ground side of the attenuator. The calibration signal is provide by the CALB DAC in the DEP. CALA and CALB can be utilized together to measure command mode rejection in TVMDC. The TVMCALN input can also be used to offset the TVMDC preamp, although this mode will not normally be used.
- TVMEN - Indicates the state of the high voltage, high vacuum momentary (N.C.) relay which connects the tether to the TVM attenuator.
- TVMFC - Solid state relay which compensates for peaking at 300 kHz in the x1 gain mode of TVMDC.
- TVMG1 - Software derived indicator for the TVMDC preamp gain.
- TVMG2 - Software derived indicator for the TVMDC preamp gain.
- CMSDOL - The CMSDOL bit is hardware derived and indicates the presence of a high bit at the input to the CMS AND gate. The CMSDOL is provided by the SFMDM. The DOL voltage is acquired in the DEP through the analog mux as an analog signal. The CMSDOL bit is actually the msb of the digitized signal.
- SHUNTDOL - Hardware bit for SHUNT.

TCVMSW5 (18,13) Fifth byte of software TCVM relay states.

- HSK0 - The lsb (bit 0) of the 4-bit analog mux address in the TCVM. All TCVM housekeeping signals are multiplexed onto one signal which is digitized in the DEP. This signal is software derived in the DEP and only indicates the bit value when directly commanded. It is normally not meaningful if the SETS TM data acquisition process is running in the DEP.
- HSK1 - Bit 1 of the mux address. See HSK0.

- HSK2 - Bit 2 of the mux address. See HSK0.
- HSK3 - Bit 3 of the mux address. See HSK0.
- HSKPEN (bit 7) - Housekeeping data from the TCVM requires 16 commands per second to set the TCVM analog mux. This process can be turned off, so that TCVM housekeeping data is lost if HSKPEN is false. HSKPEN is set by a DEP variable command.

TCMVTRLA (8,13) There are two large power resistors in parallel which form the 25 kohm load resistor. Each is potted in an aluminum housing and mounted to the base plate, one resistor on each of the long sides of the TCVM. Each housing has a temperature monitor mounted near the top middle. Since the geometry and power dissipation of the two units should be nearly equal, the two temperature monitors TCVMTRLA and TCVMTRLB are essentially redundant.

TCMVTRLB (19,13) See TCVMTRLA.

TCMVTRLC (0,10) The 250 kohm load resistor bank can dissipate as much as 100 watts at 5 kV. TCVMTRLC is the temperature of this bank of three series resistors which constitute the 250k resistor. They are potted in an aluminum housing located near the center of the TCVM, attached to the base plate between the two high power resistors of the 25k load bank. This sensor is not redundant.

TCVMM15 (0,11) Monitors the -15 volt TCVM power supply.

TCVMP15 (1,11) Monitors the TCVM +15 volt supply.

TCVMTEL (12,10) The TCVM Electronics temperature is measured in the middle of the card cage.

TCVMTPS (13,10) The Power Supply temperature is monitored at TBS on the Low Voltage Power Supply assembly.

TCVMTBP (14,10) The Base Plate temperature is measured on the bottom of the TCVM in the vicinity of the two attenuators.

TCVMTCAP (16,10) The PROT capacitor temperature monitor is located on top of the capacitor and attached to the mounting bracket.

TCVMPA (20,13) The TCVM is a sealed enclosure filled with sulfur hexafluoride (SF6) to about one atmosphere. Redundant pressure monitors with f.s. range of 15 psi should have virtually identical output values.

TCVMPB (21,13) Redundant sensor with TCVMPA.

TCVMP28 (23,13) Monitors the TCVM +28 volt power supply which supplies power to the TCVM high voltage relays. A secondary 28 volt supply internal to the TCVM is used to isolate the relays from the Orbiter power.

TCVM5 (2,11) Monitors the +5 volt power supply output in the TCVM which supplies power to the TCVM logic including the TCVM command byte decoder.

TCVMCMD (18,9) This byte is the last command sent to the TCVM, but housekeeping mux address commands are excluded. There are 16 bytes sent to the TCVM each second to acquire TCVM housekeeping mux data.

GM15 (5,11) The -15 volt monitor is set at full scale for 5 volts and consequently produces a value of 0xff for the 5 volt f.s. 8 bit digitized signal. The DEP can gain switch to 10 volts full scale for this sample which will result in a sample at half scale which is not out of range. The same treatment is used for the +15 volt monitor. The normal data acquisition process uses a gain of x2 (-g1) for sampling. In order to sample the GM15 signal within the dynamic range of the digitizer, use a direct admux command which will readout into the burst mode: `d admux b -g0 e b e b`. This command will alternately read GM15 and GP15.

GP15 (3,11) See GM15.

G5 (4,11) Monitors the 5 volt power supply for the FPEG logic circuits which are CMOS.

G120 (6,11)Each of the two FPEG GUN heads utilizes a separate power section. The power converters operate from pre-regulated 20 volts.

G220 (7,11) This is the GUN 2 pre-regulator voltage. See G120.

CCPM15 (9,11) There are six power supplies in the CCP. Two of them for each of the Charge and Current electronics are floating at the offset voltage supplied by the CDAC and QDAC. The output buffer and signal conditioning electronics are supplied by the two supplies that are monitored.

CCPP15 (8,11) Same as CCPM15.

CV (10,11) The current probe can be offset in potential with respect to the Orbiter frame in order to measure the voltage dependence of the plasma current to the Orbiter. The offset voltage is supplied by a DAC in the DEP (CDAC). CV is a hardware measure of the voltage offset.

QV (11,11) The charge probe can be offset in potential with respect to the Orbiter frame. The offset voltage is supplied by a DAC in the DEP (QDAC). QV is a hardware measure of the voltage offset. When the charge probe offset is enabled, QV can be used to produce an in-flight calibration of the charging voltage of the charge probe sensor.

IPP6 (12,11) Plus six volt supply to SRPA.

IPM6 (13,11) Minus six volt supply to SRPA.

IP10 (14,11) Magnitude of the 10kHz signal applied to the ion probe Related to the SRPA.

IP85 (15,11) Magnitude of the 8.5kHz signal applied to the ion probe. Related to the SRPA.

LPAC (18,11) Magnitude of the AC signal applied to the langmuir probe. Related to the SRPA.

LPSWP (19,11) Langmuir probe sweep voltage monitor.

VSPIB (20,11) Voltage monitor of Orbiter 28 volt power at the input to SETS.

PWRSW (0,12) Hardware byte as read from the DEP parallel input port from the SPIB. Contains the status bits for each instrument power, e.g. SPIBPWR.

- CCPPWR - Hardware derived monitor that shows 28 volts exists on the CCP side of the fuse located in the SPIB. The monitor is powered by the SPIB.
- FPEGPWR - Hardware generated bit which indicates that 28 volts exists on the FPEG side of the fuse located in the SPIB. The monitor is powered by the SPIB.
- HTRPWR - Hardware derived bit monitored in the SPIB after the fuse which indicates that 28 volts is available to the SRPA preamp heater. Current will not flow unless the temperature sensor relay is closed. This relay cannot be overridden except by placing a shorting wire across the sensor terminals.
- SRPAPWR - Hardware derived bit monitored in the SPIB after the fuse which indicates that 28 volts is available to the SRPA electronics.
- AMAGPWR - Hardware derived bit which indicates 28 volts after the AMAG fuse which is located in the SPIB.
- SPIBPWR - Hardware derived bit which indicates the presence of 28 volts at the SPIB side of the fuse. The SPIB power is switched by DOL command in the SPIB and turns on power supplies which supply power for the SPIB monitors. Unfortunately, this bit is always 1 even when the SPIB is off because the DEP pulls the bit high and the SPIB must be on to pull it low (which never happens). SRPA power on is only indicated accurately by nonzero current in the ISRPA signal.
- TCVMPWR - Indicates the TCVM power on/off state as switched in the SPIB.

CP (*,4) The CD signal is buffered by a peak hold detector which has a fast rise time and a slow decay time. CP is only significant for increased electron collection (Orbiter positive). The decay time constant is less than one second. If fast current pulses are detected, the CP value will be greater than the CD signal whenever the duration of the pulse is short compared to one second.

QP (*,5) The output of the direct charge probe signal QD is buffered by a peak hold detector of identical design to the CP signal. It has a fast rise time and a slow decay time, but decays in less than one second. A difference between the QP and QD signals indicates that the surfaces of the Orbiter are being rapidly charged and discharged. The peak detector is only significant for positive going Orbiter potentials such as produced by FPEG pulsing. The sense of the signal is positive for positive going Orbiter potentials.

IPHI (*,6) One of two channels of data from the ion probe. The data gathered is the second derivative of current to the ion probe with respect to voltage. When two successive sweeps of data from both IPHI and IPLO are combined using the correct algorithm the density and temperature of ions can be obtained.

IPLO (*,7) See IPHI.

LP (*,8) The first derivative of current to the langmuir probe as a function of voltage. Yields the density and temperature of electrons.

CD (*,14) The direct current probe output is a dc coupled, low pass filtered signal which measures the current to the large, gold plated Current sensor on top of the CCP. The current is displayed as a current flux since this quantity is physically significant and does not depend on the surface area of the sensor. There is a high gain and low gain state for the signal which is indicated by a software derived bit CCPCGAINHI and requires switching scaling factors. This switch is implemented by supplying two entries for CD and selecting one or the other of the measurement numbers to display depending on the state of CCPCGAINHI.

QD (*,15) The charge probe output is dc coupled and low pass filtered. This signal measures the charge induced on a piece of Flexible Reusable Surface Insulation (FRSI) which is from the same batch as used on the Shuttle Columbia. The charge can also be used to calculate the voltage across the dielectric insulator since the capacitance (112 pF) of the FRSI insulation is known. The Charge Probe is an indicator of the behavior of the FRSI covered surfaces on the Orbiter and provides a measure of the vehicle potential relative to the surrounding plasma. The sense of the QD charge probe signal is positive for positive going Orbiter potentials such as produced by the FPEG. When the Orbiter is negative with respect to the surrounding plasma, the signal will be negative.

QDCDLSN (*,16) Except for signals acquired by the AD0/AD1 boards, all signals digitized in the DEP are 12 bits. Most signals are truncated to the most significant byte of the 12 bits before inclusion in the tm page. In order to increase the precision of the CD and QD measurements, the Least Significant Nibble (i.e. the remaining lower 4 bits) are packed into one byte and sent as QDCDLSN. The MSByte is sufficient for realtime operations and crew display on the DDCU.

TCM (*,17-18) The Tether Current Monitor is a non-contacting measurement of the total current in the tether. The tether connection to the TCVM is carried on an insulated wire which is looped through a hall effect type current monitor five times to increase the sensitivity. This device is the same model used in the Deployer MCA. The TCM measurement is paced by the Multibus clock and is sampled with a circuit which is identical to the TVMDC digitizer except for the sample rate.

TVMDC (*,18-19) TVMDC is the primary measurement of tether voltage. It is a 16 bit sample taken 15 time per minor frame. The digitizer is paced by the Multibus clock and is a self-calibrating digitizer with very high linearity. The input TVMDC preamp has three gain states: x1, x10, and x100, which are indicated by TCVM status word bits TVMG1 and TVMG2.

TVMDCLSB (*,21-34) The DC coupled measurement of tether voltage is sampled 15 times per minor frame (360 times per second). The first sample of each 15 is sent as a 16 bit word in the TVMDC column. The LSByte of the remaining 14 samples are sent in the TVMDCLB (TVMDC Least significant Byte). The dynamic range of the tether voltage is approximately zero to - 4700 volts. The TVMDC channel has a range of +/- 10 kV. Therefore, the LSByte corresponds to about +/- 40 volts. The maximum rate of change of the tether voltage under nominal operations will be less than 5 volts/second caused by variations in the magnetic field and the trajectory. Consequently, it is possible to increase the sample rate without losing track of the value of the MS-BYte. During fast transients caused by switching current in the tether, the CMM and TVMac measurements will provide the higher time resolution. **Note:** The format of TVMDCLSB cna change depending on TVMDCMODE. See section 4.2 for complete details.

BMID (0,35-36) Id of the burst-mode packet.

BMSEQ (0,37-38) Burst-mode packet sequence number.

BMCNT (0,39,40) Number of data bytes in the burst-mode packet.

BMDATA () Data portion of the burst-mode packet.

INT1PD (23,47-50) Period of the INT1 interrupt. The period is a 24-bit number which is the concatenation of a 16-bit value and an 8-bit value. An example computation is shown below for a 24Hz interrupt.

```
Multibus clock frequency : 9.83026 Mhz
SFMDM second: .99841

1 per 1/24 second @ 9.83026 Mhz and .99841 sec.

408942.9120 = 9.83026 * .99841 / 24

cnt = (1<<22) - ( 408942.9120 - 3 )

hi = cnt >> 6          = 0xe70a
lo = cnt & 0x3f        = 0x14

INT1PD = 0xe70a14
```

TVMDCMODE (23,52) Describes the format of the TVMDCLSB data. See the section 4.2 TVMDC data below for a complete description.

FPEGONTIME (23,53) The number of seconds that the filament has been on. Starts counting at zero when the hardware bit FILON is set. It is intended to enable limit type checking of the filament to prevent the filament being left on accidently for an extended period.

CCPONTIME (23,54) CCP commands are executed by writing a byte to one of the DEP parallel ports. When these bits are high, power is applied to the CCP relay. In order to reduce stress on the CCP relays, it is important to remove power once the state has been established for the latching relays. The momentary relays should not need to be on for more than about 10 seconds in any case. This byte counts the number of seconds that a CCP relay has been on and a CCP command not issued. Therefore, it is possible to hold power on a relay and reset CCPONTIME by simply continuing to issue the same command. This byte can be limit checked.

CHKSUM (23,55-56) The two's compliment 16-bit checksum of the telemetry page. Stored as a little-endian.

ENABLES (22,13) This status byte contains the bits which show the state of the software enable/inhibit for 5 special command bits. If these bits are 0, the DEP software will inhibit the corresponding hardware command. The enable bits are set using variable type commands.

- FPEGFILEN
- FPEGHVEN
- SHUNTEN
- PROTEN
- CMSEN

GAND1 (0,13) FFPEG gun 1 anode current.

GAND2 (1,13) FPEG gun2 anode current.

GCAT1 (2,13) FPEG gun1 cathode current.

GCAT2 (3,13) FPEG gun2 cathode current.

GFOC1 (4,10) FPEG gun1 focus current.

GFOC2 (5,10) FPEG gun2 focus current.

IDEP (4,13) DEP 28 volt Orbiter current is connected directly to the DEP through the SPIB and not switched internal to SETS. The DEP comes up when SETS power is turned on at the SPCB.

IFPEG (5,13) The FPEG 28 volt Orbiter power current is monitored in the SPIB using a hall effect type sensor. Power is switched through the FPEG DOLs in the SPIB.

IHTR (10,10) The SRPA preamp mounted on the SRPA TOWER has a surface heater which is controlled by a thermostat mounted to the base of the preamp. Current is supplied from Orbiter power and monitored in the SPIB with a hall type sensor.

ISRPA (11,10) SRPA current is monitored in the SPIB with a hall type sensor. This is the 28 volt Orbiter power.

ICCP (9,10) Power to the CCP is switched in the SPIB and current monitored with a hall type sensor.

IAMAG (8,10) Monitors the current to the AMAG power supply using a hall type sensor in the SPIB. Orbiter 28 volt power.

ITCVM (7,13) The TCVM 28 volt Orbiter power input current measured with a hall effect sensor in the SPIB. The bit is acquired through a DEP parallel port bit and derived in hardware in the SPIB.

ISPIB (7,12) Current drawn by the SPIB.

MSW (2,12) The Message Status Word shows the state of the DEP software. There are five basic states. See the appendix for a list.

FLT1 (3,12) Faults detected in the DEP software execution cause an error code byte to be issued through FLT1. A value of 0 for FLT1 indicates "no fault". Multiple faults within one second will overwrite the output byte so that only the last byte written before the TM page is assembled will be contained in the telemetry downlink. However, a count of all faults is shown in FLT1CNT. The FLT1 byte will remain until it is reset by command or superceded by a later fault. The reset command for FLT1 is "`v flt1 0`".

FLT2 (4,12) The DEP can be enabled for limit checking of a few measurements which have been selected pre-mission and hard coded into the DEP software. An Out-Of-Limits (OOL) condition, when enabled, will cause a fault code byte to be written to FLT2. The FLT2 byte will remain until overwritten by a later OOL condition or reset by command. A count of OOL conditions is shown in FLT2CNT which increments once each second until the fault is corrected. The FLT2 byte can be reset by command using the variable command "v flt2 0".

FLT1CNT (14,9) This byte contains the integer number of FLT1 errors issued by the DEP.

FLT2CNT (14,11) This byte contains the integer number of FLT2 errors issued by the DEP. Unless the OOL condition is corrected, this number will continue to increment by one each second.

MFCNT (22,9) The DEP acquires a TM page at the rate of 24 minor frames per second based on an internal clock. This byte indicates the number of minor frames which were acquired during the previous 24 minor frame requests (MFRs) by the SFMDM. MFCNT is reset to zero if the current value of MFCNT¿=24 and MFR=23, following an INT1 interrupt signal. If this condition does not exist, MFCNT is incremented by one. During nominal operation, MFCNT should always be 24 to indicate that 24 minor frames were collected corresponding to 24 minor frame requests. If the INT1 rate is slower than the MFR rate, MFCNT will be around 47 and a telemetry page will be transmitted twice. If the INT1 rate is faster than the MFR rate, MFCNT will be around 25.

TSPIB1 (22,10) SPIB internal temperature monitor located at TBS.

TSPIB2 (16,9) See TSPIB1.

TSPIB3 (17,9) See TSPIB1.

FPEGCNT (14,9) A software count of the number of times a command is sent to the FPEG and completed. The count corresponds to the number of times the word enable flag in the FPEG latch has been dropped. Multiple commands sent during a second can be detected by this counter.

FPEGSW1 (19,9) A software derived status byte which contains bits indicating the state of the FPEG bits.

- FPEGENCOM - This bit must be set true for the FPEG to accept commands.
- FPEGENONTM - This bit must be set true for the FPEG to accept on time commands.
- FPEGENOFTM - This bit must be set true for the FPEG to accept off time commands.
- FPEGENPUL - This bit must be set true for the FPEG to accept number of pulse commands.
- FPEGRUN - Software bit which indicates the state of the RUN bit in the FPEG command which would be sent. Must be high for the FPEG sequencer to run.
- FPEGRUNBAR - Software bit which indicates the state of the RUNBAR bit in the FPEG command which would be sent. Must be low for the FPEG sequencer to run.
- FPEGGUN1 - The FPEG has two separate gun heads 1 and 2. Either one or both can be used. This bit indicates the software state of the command bit which would be used to send a command to the FPEG.
- FPEGGUN2 - See FPEGGUN1.

FPEGSW2 (20,9) Same as FPEGSW2 but it represents different bits.

- FPEGFIL - This bit must be set true for an FPEG command to turn on the filament. This bit is enabled by FILEN which is a variable set by ground command only. The FPEG must also be armed by installing the arming plug before power can be applied to the filament power converter. It is possible to have this bit on and still not have the filament on. The filament on/off status is not indicated by this bit, but rather it indicates the status of the filament command bit for an FPEG command.
- FPEGHV - Software state of the command bit which would turn on the high voltage power converter in the FPEG when an FPEG command is sent. This bit is software enabled in the DEP by HVEN. It does not indicate that high voltage is on or not.

- FPEGSTOP - Software bit in the FPEG command message which forces a stop to the FPEG and turns off the filament and high voltage. If a sequence is running, the STOP bit will not terminate the sequence. The beam can be turned off by simultaneously issuing one short pulse. The DEP automatically issues this set of command when given a direct FPEG STOP command.
- FPEGGO - The Go command drops the FPEG command word enable bit and latches the command into the FPEG. This bit indicates that the GO command was issued.
- FPEGSEND - The FPEG send command sends and executes a command to the FPEG using whatever the current status of the FPEG command bits is at the time.
- FPEGREADY - Software derived bit which indicates that a command has been loaded into the FPEG command register and will execute as soon as the FPEGGO command is issued to latch the command. Usage is to facilitate accurate timing of FPEG commands with data acquisition.
- FPEGFILON - Software derived bit which is set when a command is sent to the FPEG which would turn the filament on.

FPEGSW3 (1,12) A hardware derived byte which shows the status of the FPEG. The bits are sampled at the DEP parallel port.

- FILON (bit 1) - Hardware derived indicator that the filament is on. Unambiguous.
- FPEGSGP - The stretched gun pulse has a fast rise time and slow fall time which provides a bit indication that a pulse has occured sometime in the last second. It is not possible to determine the difference between a series of fast pulses and a "dc" pulse solely from the SGP. The GP coincides directly with the gun pulse and may be zero in the telemetry even though the FPEG is pulsing. In this case, the SGP will be high.
- FPEGCL - Hardware derived indicator that the FPEG beam is arcing. Because the FPEG is self-protected and recovers so quickly, this indicator bit may not be set true even if an arc-over should occur.

- FPEGSEQEN - Hareware derived bit from the FPEG which indicates that the FPEG sequencer is running. The sequencer will run without the filament or high voltage turned on. Both SGP and GP will show pulses when the sequencer runs.
  - FPEGHV (bit 5) - Hardware derived bit indicating that the FPEG high voltage converter is on. Either gun1, gun2 or both hv converters may be on.
  - FPEGHVARM - Hardware derived bit which indicates that the FPEG arming plug is installed. If this bit is on, the enable bits must not be set or the FPEG filaments will quickly burn up in air on the ground.

FPEGONTM (17,12) A software derived byte which indicates the on time command which would be sent to the FPEG in a command. It has a value of 0x00 to 0x1E and is decoded with the same lookup table as the off time byte FPEGOFFTM.

FPEGOFFTM (18,12) A software derived byte which indicates the off time command which would be sent to the FPEG in a command. It has a value of 0x00 to 0x1E. The off time is decoded from a lookup table which approximates a logarithmic table from 600 nanoseconds to 107 seconds.

FPEGNPULS (19,12) A software derived byte which indicates the number of pulses command which would be sent to the FPEG in a command. It has a value of 0x00 to 0x0F.

GUNSYNCSW (6,10) A hardware derived status byte which contains the DCORE syncs, PDECU1 and PDECU2 (which are sometimes called EGASYNC1 and EGASYNC2). It also contains the GP bit.

  - EGASYNC1 (bit 0) - Hardware measurement of the EGA sync bit.
  - EGASYNC2 (bit 1) - Hardware measurement of the EGA sync bit.
  - GP (bit 3) - The Gun Pulse is sampled once each second. It is the same signal sent to SPREE ("SPREE SYNC"). If the pulse

is longer than one second, GP will give a true indication of the gun pulse. The GP and SGP signals do not necessarily indicate that a beam is being emitted since they are derived from the pulse network driving electronics. Only if the FILON and HVON bits are asserted true is the FPEG emitting a beam. Therefore, it is possible to test the FPEG and SPREE sync interfaces, telemetry, and FPEG pulsing commands on the ground without turning on the filament and high voltage.

SRPASW (21,9) Software derived status byte containing the two bits which indicate the commanded value of the SRPA modes for level and sweep.

- SRPASWPMD (bit 4) - Software derived value for the sweep mode bit.
- SRPAACLEVEL (bit 5) - Software derived value for the aclevel mode bit.

CCPSW (20,12) A software derived status byte which indicates the commanded state of the CCP relays. Each bit has a separate entry in the database.

- CCPRELAYON - Relay status.
- CCPQGAINHI - Software derived bit which indicates the charge probe is in high gain. When this bit is set true (1), the two high gain scaling factors for QD and QP should be selected (N04V014S and N04V015S).
- CCPCGAINHI - When this bit is high, the current probe preamp is in high gain and the high gain versions of CD and CP should be displayed (N04C011S and N04C012S).
- CCPEN - Software derived bit which indicates the state of the offset enable relay. This relay connects the reference side of the current probe preamp circuits either to the CCP frame or to the CV input signal.
- CCPQEN - Software derived bit which indicates the state of the offset enable relay connected between the charge probe preamp circuit and either the CCP frame or QV.

- CCPCFILIN - A filter capacitor can be added to the feedback of the current probe preamp in order to add filtering to the signal. This bit is a software derived indicator of the state of the current probe low pass filter.
- CCPCAL - A resistor can be connected to the input side of the current probe preamp. The input is not disconnected from the current probe sensor during CCAL. If an offset is applied (CV), a current is injected into the current probe preamp equal to CV/input resistance of 220kohm.
- CCPQRESET - The drift in the charge probe preamp is caused by bias current charging the feedback capacitor. The reset command shorts the capacitor momentarily to rezero the output. When the charge probe is reset, all history of charging is lost and the output then indicates changes following the reset. The reset relay is a momentary on relay which should not be operated for more than 10 seconds nominally.

IPSWP1 (17,11) The ion probe sweep voltage IPSWP is sample at four equally spaced times each major frame. The samples are taken near the same time as the acquisition of minor frames 0,6,12, and 18. The order of the samples is IPSWP1, IPSWP2, IPSWP3, and IPSWP4, although they occur at different times and places within the tm page than when they are acquired.

IPSWP2 (23,9) See IPSWP1.

IPSWP3 (23,10) See IPSWP1.

IPSWP4 (15,10) See IPSWP1.

OMI (5,12) The Operating Mode is a byte which is decoded in a lookup table to indicate in plain language the major operating mode of the SETS instrument. The OMI and OMITTG function the same as the OCI and OCITTG bytes and are set by commands issued by DEP internal sequences.

OMITTG1 (6,12) Operates the same as OCITTG1. See OCITTG1.

OMITTG2 (6,13) See OCITTG2.

OCI (8,12) The Operating Cycle is decoded from a lookup table to indicate what FO or JFO is currently being executed by the DEP. The decoded phrase is descriptive of the JFO and the corresponding OCI Time-To-Go indicates the number of seconds remaining. The OCI is a DEP variable which is set by a variable command. These commands are embedded in the DEP internal sequences along with the integer number of seconds for OCITTG.

OCITTG1 (9,12) The OCI Time-To-Go is a 16 bit word with the MSByte in OCITTG1 and the LSByte in OCITTG2. This 16 bit word is decremented by the DEP each second and shows the number of seconds remaining before the end of the present Operating Cycle. A value of zero indicates a static condition in which the Operating Cycle shown by the OCI byte has terminated. The OCI value is not reset at zero seconds.

OCITTG2 (9,13) This is the LSByte of the 16 bit Time-to-Go for OCI. See OCITTG1.

SOCI (11,12) The SOCI byte is decoded from a lookup table and indicates a finer resolution of the SETS DEP sequence. SOCI and SOCITTG are used in the same way that OCI and OCITTG are used.

SOCITTG1 (12,12) See OCITTG1.

SOCITTG2 (12,13) See OCITTG2.

STP (14,12) The STP is not decoded from a lookup table. The decimal value of STP indicates the STEP number of the currently executing FO. STP is the most rapidly varying of the four indicators: OCI, SOCI, OMI, and STP. It also has a TTG word.

STPTTG1 (15,12) See OCITTG1.

STPTTG2 (15,13) See OCITTG2.

TGGH1 (10,12) The two gun heads 1 and 2 have separate power converters. The "Gun Head" temperatures are not actually measured at the gun head, but rather at the power converter which drives the FPEG gun head filaments.

TGGH2 (10,13) See TGGH1.

TGEL (11,13) The FPEG electronics temperature is monitored on one of the electronics boards in the FPEG card cage.

GHV1 (13,12) Monitors the electron beam accelerating voltage for FPEG GUN head 1.

GHV2 (13,13) Monitors the electron beam accelerating voltage for FPEG GUN head 2.

TCB (17,10) The CCP Box temperature is monitored on one of the pc boards in the CCP card cage where the command driver and output buffer circuits are located.

TCP (18,10) The Current Plate temperature is monitored on a standoff attached directly to the Current Plate side of the CCP and indicates the surface temperature of the gold plated current probe.

TQE (19,10) This sensor is attached directly to the Charge probe preamp electrometer. Because the input bias current doubles with each 10 DEGC rise in temperature and produces an increased drift in the QD and QP signals, this monitor provides a means of backing out the drift. The preamp has been changed for TSS-1 to a new very low bias current type which markedly improves the drift behavior at room temperature and above. For the operating temperatures on orbit, the drift even at high gain should be neglibible. Some warming of the preamp is detected at power on due to self heating.

TIPA (20,10) Temperature of the SRPA electronics.

TIPP (21,10) Temperature at the base of the probe (SRPA).

TDEP1 (1,10) DEP temperature measured on the AD0 electronics card near the center of the DEP.

TDEP2 (7,10) DEP temperature measured on the AD1 electronics card adjacent to the TDEP1 measurement. Temperature difference between TDEP1 and TDEP2 is on the order of 2 DEGC.

GDQ1 (2,10) FPEG gun 1 emitted charge (integrated current) during sample window. This channel saturates for gun emissions ¿ 100 microseconds.

GDQ2 (3,10) Same GDQ1 but for FPEG gun2.

BX (21,12) The AMAG X axis is directed positive downward into the payload bay, parallel to the normal Orbiter Z axis. BX is the same signal source as SBX except sampled one half second earlier. BX is sampled in minor frame 0 and SBX is sampled in minor frame 12.

BY (22,12) The AMAG Yaxis is directed positive toward the Orbiter tail, which is the normal Orbiter +X axis. BY is the same signal as SBY but sampled at the same time as BX. SBY is sampled at the same time as SBX.

BZ (23,12) The AMAG Z axis completes a right hand system and is directed towards the Orbiter port wing (normal orbiter-Y axis). It is sampled at the same time as BX, BY and is the same signal as SBZ except for the sample time.

SBX (21,11) SBX, SBY, and SBZ are captured by the SFMDM and transmitted to SPREE. BX,BY, and BZ are the values sent to the DDCU. The signals are the same except for a half second difference in the sample time. On the ground, the two groups should be treated as one signal with twice the sample rate.

SBY (22,11) See BY and SBX.

SBZ (23,11) See BZ and SBX.

4.1 Time

Time is represented by three fields in the telemetry: GMT, CPUTIME and MBUSTIME. GMT is the Greenwich Mean Time. CPUTIME is the internal time of the master cpu. The 32-bit number represents the number of one millisecond ticks that have elapsed since the cpu was started. MBUSTIME is the 48-bit multibus time maintained by the sdio board. MBUSTIME and CPUTIME are recorded for telemetry when the GMT changes. The MBUSTIME (kept by a counter on the sdio board) is saved when a new GMT second is received from the SFMDM. When the cpu fetches the GMT and multibus time from the sdio board, it saves the current cpu time.

4.2 TVMDC

The TVMDC signal is nominally sampled at 360Hz, however, there is not sufficient bandwidth for transmission of all 16-bit quantities at this sampling rate. The TVMDC section of the telemetry page consists of 24Hz 16-bit data (every 15th sample of the 360 samples). The TVMDCLSB section of the telemetry contains a compact version of the remainder of the data. Because of concern due to noise levels and signal characteristics, several methods of compacting the full rate are supported. The format of the TVMDCLSB data is given by the TVMDCMODE byte. The values of this byte and the formats are explained below.

- `00-0F` : If the value of TVMDCMODE is zero, the TVMDCLSB data consists of the least significant bytes (8-bits) of the other 336 samples. If the value of TVMDCMODE is greater than zero and less than 10, the least significant nibble (4-bits) is used as a shift value, i.e., the other 336 samples are shifted X bits to the right (divided by 2^X) before being transmitted. For example if TVMDCMODE is 14, the samples are shifted right 4 bits (divide by 16). This shifting functionality is supported if the noise level is high. During reconstruction of this data it is important to take into account rollover. For example, if two adjacent values are F4 and 04, the most significant byte must be incremented to account for the possibility of a rollover. Unfortunately, there is no way to be sure that a rollover occurred.

- `10-1F` : If the value of TVMDCMODE is in this range, TVMDCLSB contains the differences between adjacent samples. As above, the lower nibble of the TVMDCMODE value indicates the number of bits the original samples are shifted right. In this case however, the samples are shifted right before the differences are computed. If a difference is greater than 127 or less than -127, 127 or -127 is transmitted, respectively. The actual difference, however, is stored and the remainder ($actual - 127$) is propagated to the next difference computation. This allows full reconstruction of the data even when some adjacent differences are greater than 127. This mode is useful if the data is changing rapidly.

- 20 : If TVMDCMODE is 20, the next 7 samples of the after each 15th are placed in the TVMDCLSB data section. The 7 16-bit samples are stored in big-endian byte order.

Other values of the TVMDCMODE are invalid.

4.3 Burst-mode

The burst-mode section of the telemetry is 518 bytes in length. The first six bytes are header and the remaining 512 bytes are data. The format of the header is:

- 2 bytes (little-endian short) representing the packet id.
- 2 bytes (little-endian short) representing the packet sequence number.
- 2 bytes (little-endian short) representing the number of bytes in the data portion of the packet. The maximum is 512.

The packet id represents a logical stream of data. The sequence number determines which packet in the stream it is. Packets for a stream are guaranteed to be transmitted in order. The id must be greater than zero. Packets with an id of zero denote a packets with no data. These are filler packets. Certain ids have been set aside for special purposes. In general, ids less than 256 represent an ascii stream. Ids greater than or equal to 256 represent digitizer or other data. Fixed ids for ascii streams are listed in the appendix. The display program must know the type of data in a stream. A new stream is implicitly started when the sequence id of a packet is zero.

If the burst-mode stream is data from the AD0/1 boards, the first packet in the stream (sequence number 0) will contain a header describing the data. The format of this header is given below.

```
long  count;                /* number of bytes in stream
                               including header and data */

short ad_adnum;             /* which device 0,1,2,3 */
short ad_id;                /* experiment id */
long ad_gmt;                /* gmt of sample */
short ad_cputime[3];        /* cpu time of sample */
short ad_samplecmd;         /*_sample command code */

long ad_period;             /* sample period */
long ad_wait;               /* wait period */
short ad_trigger;           /* trigger mode */
short ad_numsamples;        /* number of samples taken */
short ad_ramaddr;           /* start ram address */

short ad_flags;             /* adpac mode flags */

char _tcvmsw1;              /* tcvm relay status */
char _tcvmsw2;
char _tcvmsw3;
char _tcvmsw4;
char _tcvmsw5;
char _enables;              /* software enable flags */
char _fpegsw1;              /* fpeg status */
char _fpegsw2;
char _fpegsw3;
char _gunsyncsw;            /* gun sync status */
```

All variables are little-endian byte order. The number of data bytes is determined by subtracting the size of the header structure from the value of the count field.

Note: The data for the 8-bit digitizers (1 and 3) is byte swapped.

5 Appendix

5.1 Burst mode ids

- `10(0xa)` - standard output (command echo, error messages, etc.)
- `11(0xb)` - output of limit checking process (not `lim` commands), e.g., limit checking faults.
- `12(0xc)` - output of timeline commands
- `13(0xd)` - output of limit table commands
- `14(0xe)` - output of variable commands
- `15(0xf)` - output of parameter commands

5.2 Process priorities

- `-2` - Housekeeping process
- `-1` - Command process
- `0` - Time process
- `0` - Limit checking process
- `3` - Application processes
- `4` - Send processes (send, ad send, ad file commands)

5.3 Debug flags

- `0` - Debug off
- `nonzero` - Debug on
- `0x10` - AD pac housekeeping debug. Check for data dropouts and report them. This flag is reset after such an error occurs. It must be set again manually.

- 0x20 - INT1 debug. Not used.
- 0x40 - SDIO debug. If set and limit checking for sdio is enabled, the elapsed time since a minor frame request was received is printed.
- 0x80 - GMT debug. If set and limit checking for gmt is enabled, the elapsed time since an int1 interrupt and a new gmt was recieved is printed.

5.4 DEP fault codes

- 0x01 - Bad logical
- 0x02 - Command error
- 0x03 - DAC command error
- 0x04 - PIOB command error
- 0x05 - ADMUX command error
- 0x06 - CCP command error
- 0x07 - FPEG command error
- 0x08 - SRPA command error
- 0x09 - TCVM command error
- 0x0a - Timeline queue full
- 0x0b - Timeline command error
- 0x0c - Sequence command error
- 0x0d - Initialize command error
- 0x0e - Rcmd command error
- 0x0f - Load command error
- 0x10 - Send command error

- `0x11` - Process fork error
- `0x12` - Exec error
- `0x13` - Parameter command error
- `0x14` - ADPAC command error
- `0x15` - SDIO error
- `0x16` - GMT error
- `0x17` - Variable command error
- `0x18` - Limit command error
- `0x19` - No memory
- `0x1a` - Housekeeping process error
- `0x1b` - Cannot open tmpage file

5.5 DEP status word

- 0 - Waiting for a command
- 1 - In a sequence
- 2 - Load to a file
- 3 - Executing a command
- 4 - Wait command

5.6 SETS OS error codes

- `EOS 0x01` - operating system error
- `EMAGIC 0x100` - bad magic number
- `ENOENT 0x101` - no entity (file usually)

- EAGAIN 0x102 - try again, insufficient resources
- ENOMEM 0x103 - no memory
- ENOTDIR 0x104 - not directory
- EACESS 0x105 - no access
- EMFILE 0x106 - max number of files
- EBADFS 0x107 - bad file system
- EBINOD 0x108 - bad inode
- ELOCK 0x109 - resource is locked
- EISDIR 0x10a - file is dir
- ENOFS 0x10b - not a filesystem
- EINVAL 0x10c - invalid argument
- EEXIST 0x10d - file exists
- ENOSPC 0x10e - no space
- ENTOLG 0x10f - name too long
- ENEMPTY 0x110 - not empty
- EPERM 0x111 - no permission
- ETIMOUT 0x112 - timeout
- ECHILD 0x113 - child
- EWBLOCK 0x114 - would block
- EXDEV 0x115 - files are on different file systems

5.7 Version 1.1 Update

The following changes have been made to the SETS application for Version 1.1.

- Add version string. The version and a copyright message prints out before the initialization of devices and processes.
- Fix a bug that causes the system to lock up if the top command in the timeline is being held.
- If `flt1` or `flt2` are set by the variable (`v`) command to nonzero values, a depfault message will be issued. Setting the flt code to a nonzero value is not an error but simply a mechanism for generating a depfault message.
- When a command is received, if the proper bit in the debug variable is set (see below), the checksum of the command will be verified. If the checksum is incorrect, a flt1 message will be issued.
- When a minor frame dropout is detected, an `mfdp` command is issued. The `mfdp` commands causes a flt1 message to be issued.
- The `soci` and `socittg` variable commands have been removed. The corresponding telemetry fields have been removed too. `socittg1` and `socittg2` have been replaced with `mfr0time0` and `mfr0time1`, respectively. `mfr0time` is the the cpu time at which the request for minor frame 0 was received.
- Fix timeline bug that causes a timeline command to be lost if it has the same time as the top timeline command.
- Replace the `int1pd` with the `mfcnt0time` which is the cpu time at which the first int1pd for a new telemetry page (also the approximate time at which the samples for the 1Hz data are taken and the first samples for the 24Hz data).
- Fix construction of adpac burst header so that the relay status reflects the current state of the relays after the relay part of an `int1` command.

New commands

`i exit (cpuid) (passwd)`

Exit the SETS application. See section 3.10 for more information on initialization commands.

`mfdp (mf1) (mf2)`

This command is an internal dep command and is not meant to be issued externally. It is used to denote a missed minor frame request. It causes a depfault to be issued.

New debug flags

- `0x100` - Command checksum debug. If this flag is set, the checksum of commands from the sdio is verified. A depfault is generated if there is a discrepancy.

New depfault codes

- `0x1c` - Minor frame request was missed.
- `0x1d` - Command checksum error.

New telemetry page The format of the telemetry page is shown in tables 6, 7 and 8.

Each of the new telemetry fields are briefly described below. Next to each variable name is its position in the telemetry page. The position is given as (x,y) where x is the minor frame (beginning with 0) and y is the column (beginning with 1).

MFR0TIME0 (12,12) Lower byte of cpu time (in milliseconds) at which the request for minor frame 0 was received.

MFR0TIME1 (12,13) 2nd byte of cpu time (in milliseconds) at which the request for minor frame 0 was received.

MFCNT0TIME (23,47-50) Lower 4 bytes of cpu time (in milliseconds) at which the telemetry page acquisition cycle is started. This is the approximate time at which the 1Hz data samples are taken and the first of the 24Hz data samples are taken.

1	2	3	4	5	6	7	8	9	10	11	12
C0	3F	0	CP	QP	IPHI	IPLO	LP	GMT1	TCVMTRLC	TCVMM15	PWRSW
C0	3F	1	-	-	-	-	-	GMT2	TDEP1	TCVMP15	FPEGSW3
C0	3F	2	-	-	-	-	-	GMT3	GDQ1	TCVM5	MSW
C0	3F	3	-	-	-	-	-	GMT4	GDQ2	GP15	FLT1
C0	3F	4	-	-	-	-	-	MBUS1	GFOC1	G5	FLT2
C0	3F	5	-	-	-	-	-	MBUS2	GFOC2	GM15	OMI
C0	3F	6	-	-	-	-	-	MBUS3	GUNSYNCSW	G120	OMITTG1
C0	3F	7	-	-	-	-	-	MBUS4	TDEP2	G220	ISPIB
C0	3F	8	-	-	-	-	-	MBUS5	IAMAG	CCPP15	OCI
C0	3F	9	-	-	-	-	-	MBUS6	ICCP	CCPM15	OCITTG1
C0	3F	10	-	-	-	-	-	CPUTIME1	IHTR	CV	TGGH1
C0	3F	11	-	-	-	-	-	CPUTIME2	ISRPA	QV	UNUSED
C0	3F	12	-	-	-	-	-	CPUTIME3	TCVMTEL	IPP6	MFR0TIME0
C0	3F	13	-	-	-	-	-	CPUTIME4	TCVMTPS	IPM6	GHV1
C0	3F	14	-	-	-	-	-	FLT1CNT	TCVMTBP	IP10	STP
C0	3F	15	-	-	-	-	-	FPEGCNT	IPSWP4	FLT2CNT	STPTTG1
C0	3F	16	-	-	-	-	-	TSPIB2	TCVMTCAP	IP85	TCVMSW2
C0	3F	17	-	-	-	-	-	TSPIB3	TCB	IPSWP1	FPEGONTM
C0	3F	18	-	-	-	-	-	TCVMCMD	TCP	LPAC	FPEGOFFTM
C0	3F	19	-	-	-	-	-	FPEGSW1	TQE	LPSWP	FPEGNPULS
C0	3F	20	-	-	-	-	-	FPEGSW2	TIPA	VSPIB	CCPSW
C0	3F	21	-	-	-	-	-	SRPASW	TIPP	SBX	BX
C0	3F	22	-	-	-	-	-	MFCNT	TSPIB1	SBY	BY
C0	3F	23	-	-	-	-	-	IPSWP2	IPSWP3	SBZ	BZ

Table 6: Telemetry Page (1 of 3)

13	14	15	16	17	18	19	20	21 - 34
GAND1	CD	QD	QDCDLSN	TCM		TVMDC		TVMDC LSB
GAND2	-	-	-	MSB	LSB	MSB	LSB	-
GCAT1	-	-	-	-	-	-	-	-
GCAT2	-	-	-	-	-	-	-	-
IDEP	-	-	-	-	-	-	-	-
IFPEG	-	-	-	-	-	-	-	-
OMITTG2	-	-	-	-	-	-	-	-
ITCVM	-	-	-	-	-	-	-	-
TCVMTRLA	-	-	-	-	-	-	-	-
OCITTG2	-	-	-	-	-	-	-	-
TGGH2	-	-	-	-	-	-	-	-
TGEL	-	-	-	-	-	-	-	-
MFR0TIME1	-	-	-	-	-	-	-	-
GHV2	-	-	-	-	-	-	-	-
TCVMSW1	-	-	-	-	-	-	-	-
STPTTG2	-	-	-	-	-	-	-	-
TCVMSW3	-	-	-	-	-	-	-	-
TCVMSW4	-	-	-	-	-	-	-	-
TCVMSW5	-	-	-	-	-	-	-	-
TCVMTRLB	-	-	-	-	-	-	-	-
TCVMPA	-	-	-	-	-	-	-	-
TCVMPB	-	-	-	-	-	-	-	-
ENABLES	-	-	-	-	-	-	-	-
TCVMP28	-	-	-	-	-	-	-	-

Table 7: Telemetry Page (2 of 3)

35 - 36	37 - 38	39 - 40	41 - 46	47 - 50	51	52	53	54	55 -
BM ID	BM SEQ	BM CNT							
BM DATA (512 bytes)									
				MFCNT0TIME	-	TVMDCMODE	FPEGONTIME	CCPONTIME	CHK

Table 8: Telemetry Page (3 of 3)

Add