PSD SEM User Manual

Version 1.0 (6/4/2000)

Author : R. Sterling (SSL), J. Knödlseder (CESR)

1. SCOPE OF THE DOCUMENT

This document is the PSD simulator user's manual. It describes the capabilities of the simulator and is the installation manual, the user guide and the software manual.

2. REFERENCE DOCUMENTS

RD1 SPI-MU-423-2493-BERK

3. APPLICABLE DOCUMENTS

AD1

4. OVERVIEW

The PSD SEM consists of hardware and software designed to simulate the logical operation of the PSD sub-assembly of the SPI telescope. The package includes hardware with the actual PSD Digital Signal Processing (DSP) unit, a Personal Computer (PC) which acts as memory spy of the PSD SEM, and PC software which allows the user to monitor and log the PSD SEM operations.

4.1. Hardware overview

The PSD SEM hardware consists of

- **A PSD DSP electronics aluminium box.** This box has a complete array of connectors to interface to the DPE and DFEE sub-assemblies, however only the main DPE High-Speed (labelled DPEA HS) and main DPE Low-Speed (labelled DPEA LS) connectors are to be used in the SEM. The PSD DSP electronics box is connected to a Personal Computer (PC) via a 50 conductor ribbon cable which divides into 40 lines for memory access and 10 lines for PSD DSP power supply. The 40 memory lines are connected on the PC side to an IDE controller interface. The 10 power lines are connected to the PC power supply. On the PSD DSP electronics box side, the 40 memory lines are connected to the DSP memory board, while the 10 power lines end in a DB9 connector for external PSD DSP electronics power supply.
- **Personal Computer (PC).** The PSD DSP electronics box is connected to a PC for power supply and memory survey. Memory survey is accomplished by connecting the DSP memory board via a 40 lines cable to the PC IDE interface. On the PSD DSP side, the cable is connected to an IO port of the PSD processor that has limited control over the processor. In addition it allows for DMA (Direct Memory Access) to the PSD memory for both reading and writing. To the PC, the PSD DSP electronics appears as a series of IO ports, located at the addresses `0x170` to `0x17F`. The ports must not be used by any other device in the PC, as they would conflict with the operation of the interface board. In particular, the IO addresses include the addresses of the standard secondary IDE controller, which is often included on modern mother boards, and which should be disabled in ROM BIOS set-up for compatibility. A VGA colour monitor is supplied with the PC that allows the visualisation of the DSP memory. A keyboard attached to the PC allows changing of the monitoring set-up and provides limited control over the PSD DSP electronics.

4.2. Software overview

The PSD SEM software consists of

- **PSD DSP software.** This software emulates the functional interfaces of the PSD sub-assembly. It is derived from but not identical to the PSD EM/FM software. The software is burned into EEPROM in the PSD DSP electronics box and can be changed by the procedure described in section 7.
- **PC memory survey software.** A software executable called SEMGSE.EXE is provided with the SEM PC which allows monitoring, logging and manipulating of the PSD DSP electronics.

5. PSD DSP electronics

5.1. Architecture

The PSD DSP electronics consists of two electronic boards. They contain a digital signal processing unit DSP32C identical to the EM/FM units, 32 kBytes of EEPROM containing the PSD DSP software, and 512 kBytes of RAM which is used for code execution and parameter storage. Additional 3 x 2 kBytes of RAM are available within the DSP32C processor, and are used for variable storage.

The memory organisation of the PSD SEM is depicted in Fig. 1. The address space is separated in a slow (x wait states) and a fast (1 wait state) memory area. Due to a particular addressing scheme, the 512 kBytes of RAM are accessible through 3 different address areas :

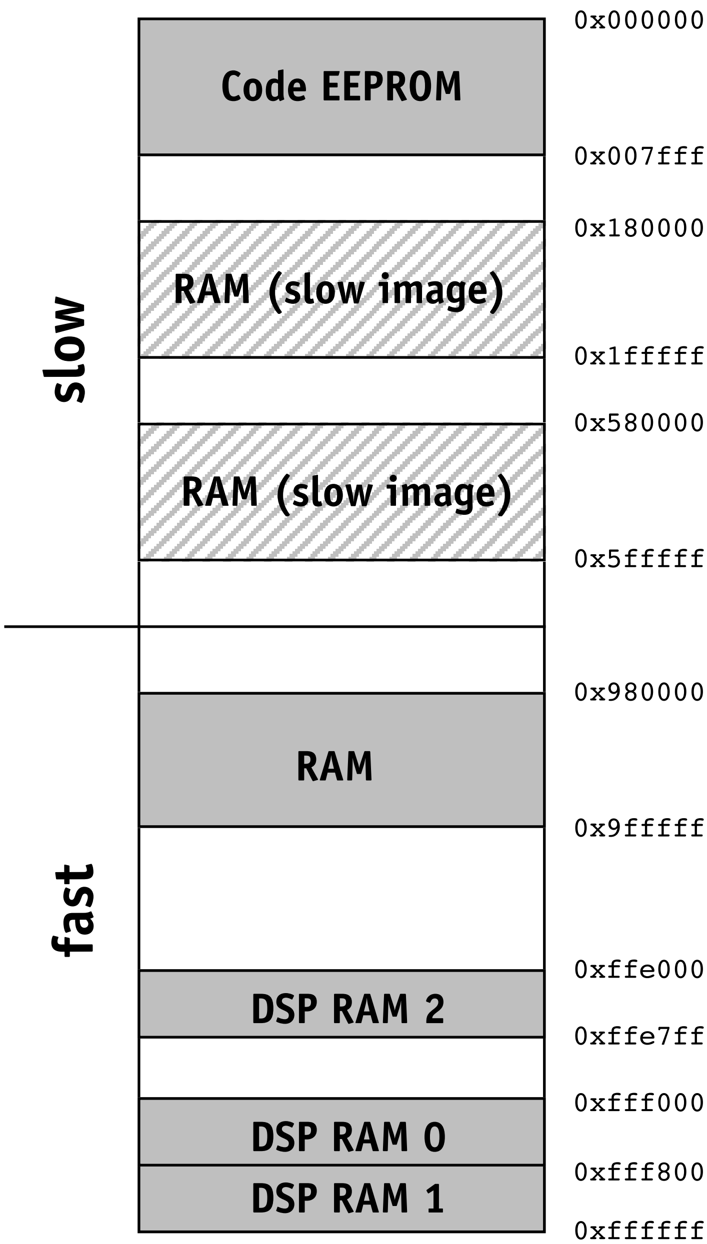

Fig. 1 : PSD SEM memory organisation.

6. SEMGSE software

6.1. Architecture

The SEMGSE software on the PC is the main executable that allows for limited control over the PSD DSP unit and that provides monitoring of the DSP memory. Note that the PSD DSP starts working as soon as the PC is switched-on, independently of the execution of the SEMGSE software.

After start-up an initialisation, the SEMGSE runs through a continuos sequence of checking for :

- a new keyboard command from the user (see section 6.2)
- new housekeeping values (read and displayed cyclically once a second)
- a new command in the PSD (indicated by a command flag raised specifically for the SEMGSE software by the PSD DSP software)
- change in the ACK / NACK state of the PSD DSP

6.2. Command list and displays

The following list resumes all commands of the SEMGSE software. A command is entered by typing the `CMD` key on the keyboard.

CMD	Action
`F1`	**Display help screen.** This screen provides a list of most SEMGSE commands (some commands are missing in this list; however they are listed in this table). Press any key to switch back to the previous display (i.e. the display from which the help screen was entered).
`F2`	**Display main screen.** The main screen is composed of 5 windows which are distinguished by their colours : 1. Dark blue (top-left) : Displays the first 10 events in the PSD event buffer. The left column contains the event ID word, the right column contains the PSD science data word (see also Fig. 2). All data are displayed in hexadecimal format. 2. Red (top-centre) : Displays the 80 words that compose a single PSD curve (the same curve is always sent through the HSL). The first word is found in the top-left corner, the last in the bottom-right corner. Words are ordered row by row. Only the 9 least significant Bits of each word are used in the HSL transfer, although the display may show any 16 Bit value. All words are displayed in hexadecimal format. 3. Purple (top-right) : Displays 11 housekeeping parameter words in hexadecimal format. For the `Status` word the PSD mode is translated into STDBY, CONF, OPER, DIAG, or CALIB. 4. Green (bottom-left) : Displays the PSD LSL response together with the time of sending (note that the time is attributed by the PC). 5. Light blue (bottom-right) : Displays the PSD LSL input buffer to visualise the DPE command sent to the PSD. A command counter is incremented after a new command has been received.
`F3`	**Display all Housekeeping data screen.** This screen shows all housekeeping data that are transmitted by the PSD. The following parameters appear : 1. **A/D+CMD :** Corresponds to Bytes 3-26 of HK12 (hex) of the Annex 21.4 interface definition (12 words). 2. **BLK13 :** Corresponds to Bytes 3-26 of HK13 (hex) and Byte 3 of HK14 (hex) of the Annex 21.4 interface definition (26 Bytes). 3. **SELSTATS :** Corresponds to Bytes 5-26 of HK14 (hex) and Bytes 3-18 of HK15 (hex) of the Annex 21.4 interface definition (19 words)

	4. **RATEHIST :** Corresponds to Bytes 19-26 of HK14 (hex), Bytes 3-26 of HK16 (hex), Bytes 3-26 of HK17 (hex), Bytes 3-26 of HK18 (hex), Bytes 3-26 of HK19 (hex), and Bytes 3-26 of HK1A (hex) of the Annex 21.4 interface definition (64 words) 5. **LIBSTATS :** Corresponds to Bytes 3-26 of HK1B (hex) and Bytes 3-16 of HK1C (hex) of the Annex 21.4 interface definition (19 words) 6. **MISC :** The first 5 words correspond to Bytes 17-26 of HK1C (hex) of the Annex 21.4 interface definition, the remaining 2 words are arbitrary and should not be considered.
`F4`	**Display PSD memory screen.** Displays 256 Bytes of PSD DSP memory. The start address of the memory block to be displayed is selected by the command `A` (see below). Note that the memory content cannot be modified.
`F5`	**Display PSD HSL buffer screen.** Displays the entire PSD HSL buffer (308 words) that is transferred on the High Speed Link to the DPE (all words are displayed in hexadecimal format). The buffer content is divided using colour codes into the following segments : 1. Green : 3 SOP words of value `0002.` 2. Dark blue : 1 word containing in the MSB (left two digits) the number of events in the HSL buffer (default : `19` hex) and in the LSB (right two digits) the number of curves in the HSL buffer (default : `5` hex). 3. Light blue : Events, 2 words each. The first word is the event ID, the second word the PSD science data (see also Fig. 2). If also curve data are present for an event, the PSD scientific data field is `0000`. As default, there are 50 event words (i.e. 25 events). 4. Purple : Curves, 48 words each. The first three words of each curve is the event ID, a blank field (`0000`) and the PSD scientific data. The remaining 45 words are the compressed curve data, which are build of 80 values of 9 Bit each. 5. Red : 3 EOP words (containing the length of the xxx; as default the EOP is `0123` hex) 6. Grey : Empty part of transfer buffer, filled with zeros. 7. Green : 3 EOT words (containing the length of the xxx, including the 3 SOP and the 3 EOP words; as default the EOT is `0129` hex)
`F6`	Optional window for miscellaneous information (not used in the actual version).
`F8`	**Display entire HK data screen.** Display the entire PSD Housekeeping data buffer (174 words). The PSD HK data buffer ; total length of HK data : 132 words
`F9`	**Append screen snap shot to log file.** Copies the actual content of the screen into the log file.
`A`	**Select memory address.** Allows to specify the start address of the memory block that is displayed in screen `F4`.
`L`	**Load pulse or event buffer from disk file.** Allows to load a binary file from disk into the event or pulse buffer of the PSD DSP memory (see section xxx for a detailed description of the binary file format). The event buffer contains the definition of the event identifiers and PSD scientific data words (250 words maximum). The pulse (or curve) buffer contains the specification of a PSD pulse shape (80 words). After typing `L` the user is asked for the type of data that should be loaded. Type `P` if you want to load a binary file into the pulse (or curve) buffer, type `E` if you want to load a binary file into the event buffer. After typing `P` or `E` the user is asked for the filename of the binary file. Specify the filename relative to the actual working directory (i.e. the directory where the SEMGSE executable resides).
`P`	**Update pulse (curve) data in main screen.** Forces update of the PSD curve data display.

E	**Update event data in main screen.** Forces update of the PSD event data display.
H	**Update housekeeping display.** Forces update of the housekeeping display.
R	**Reset the PSD DSP processor.**
F	**Fill pulse buffer with fake values.** Fills the PSD DSP pulse (or curve) buffer with some fake values. After start-up of SEMGSE, the fake values are comprised of 80 incrementing values, starting at 0 for the first value, and ending with 79 for the last value. However, if `X` is used to poke a particular value in the pulse (or curve) buffer, the fake values are replaced by the actual curve that is present in the PSD DSP pulse buffer.
X	**Poke a value into the pulse buffer.** Allows to modify one of the 80 individual values of the PSD DSP pulse (or curve) buffer. The user will be prompted for the offset (i.e. the index of the curve value) and the actual value. The offset should be specified as decimal value, comprised between 0 and 79. The value to be poked should be specified as hexadecimal value, comprised between `0000` and `FFFF`. Note that only the 9 least significant Bits of a curve value are relevant (although you will find all 16 Bits in the display). Example : Entering '12 `FF`' will replace the 13th curve value by `00FF` (note that the 1st curve value has an offset of 0).
Y	**Poke a value into the event buffer.** Allows to modify the first 20 words in the PSD DSP event buffer individually. The event buffer contains up to 125 events, each composed of 2 words. The first word of an event specifies the event identifier, the second word gives the PSD science data (see Fig. 2). The user will be prompted for the offset (i.e. the index of the word in the event buffer) and the actual value. The offset should be specified as decimal value, comprised between 0 and 19. The value to be poked should be specified as hexadecimal value, comprised between `0000` and `FFFF`. Example : Entering '4 `0000`' will replace the event identifier of the 3rd event by `0000` (note that the third event is located at the offsets 4-5).
Q	**Quit the SEMGSE program.** The user will be prompted to confirm by Y if he really wants to quit the program. The log file will be automatically closed when quitting. Note that quitting the SEMGSE program does not stop the PSD SEM.

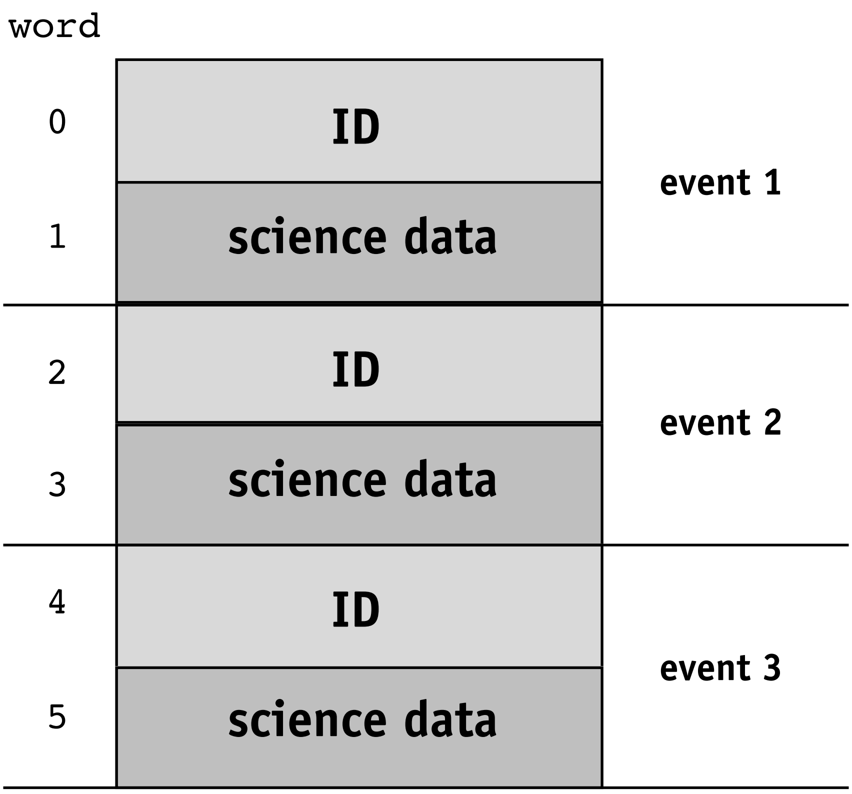

Fig. 2 : PSD DSP event buffer

5.3. Getting started

In the following a simple sample session is outlined that may help the user to get familiar with the PSD SEM system :

1. Connect the PC and the monitor to power and switch them on. Since the DSP electronics is power by the PC this box is also switched on immediately.
2. When starting the PSD SEM the first time please make sure that the following files exist in the INTEGRAL repository of the PC (operate in MS-DOS or in a DOS shell under WINDOWS) :

`SEMGSE.EXE`	The main executable
`HEXED.EXE`	A binary file editor to manipulate event lists or curves
`HEXED.DOC`	The documentation of the binary file editor
`xxx`	A VGA display driver
`EVENT.XXX`	A sample event list binary file
`PULSE.XXX`	A sample curve binary file

You need at least `SEMGSE.EXE` and xxx to run the memory monitor.

3. Invoke the SEMGSE program by typing : `SEMGSE`. A coloured full-screen DOS window should now open that shows the main SEMGSE display.
4. Type `F1` to enter the help screen which summarises all SEMGSE commands. Walk through the various SEMGSE displays by typing

`F2`	to go back to the main display
`F3`	to show the housekeeping display
`F4`	to show portions of the DSP memory (you may choose any memory

	address by the `A` command)
`F5`	to show the 308 words of data that are sent through the HSL
`F8`	to display the raw housekeeping buffers in memory
`F9`	to make a screen dump of the actual screen in the log file
`F2`	to switch back to the main display; note the log file name in the upper left of the display (you may visualise this log file after the session)
`Q`	to quit SEMGSE
`EDIT xxx.log`	to open the log file in a editor, where xxx is the name of the log file you noted above; you should for example find in the log file the screen you dumped by typing `F9` ...

5.3. Load event lists

5.4. Load curve

5.5. Modifying the SEMGSE code

6. SEM DSP software update

6.1. Introduction

In order to facilitate software updates on the PSD SEM, the ROMs U24, U26, U28, and U30 on the PSD memory board have been replaced by EEPROMs. The PSD memory, and hence also the EEPROMs, can be accessed via a "DSP Debugging Port" that is connected via an IDE interface to the PC. Software uploads can then easily be performed by programming the EEPROMs via this IDE interface.

6.2. Software preparation

The PSD DSP32C software consists of a loader program of length `0x88` (`136` Bytes) starting at address `0x000000`, followed by the functional and scientific software of arbitrary length, starting at address `0x000088`. The task of the loader program is to transfer the DSP32C software during start-up from the EEPROM to RAM. Since the DSP32C processor starts executing from address `0x000000` at reset, it is necessary that this loader also starts at address `0x000000`.

For the moment, it is assumed that the loader software already resides in the EEPROM and that only the PSD functional and scientific software should be replaced (see below for replacing also the loader). The EEPROMs are organised in memory pages of `256` Bytes each, and some time delay has to be added after programming a page to allow completion of the internal programming procedure. For this reason, the code will be split in junks of `256` Bytes (at maximum) and each junk is written to a EEPROM page separately. Code splitting is performed using the MS-DOS executable `DSPSPLIT`. This executable splits a binary file into a number of binary images. The first image has a length of `256–136=120` Bytes since the loader part of the code already resides in memory. All following images have a length of `256` Bytes.

6.2. Software programming

Several jumpers on the PSD memory board have to be changed before, during, and after the programming procedure. The concerned jumpers are `JP4`, `JP5`, and `VCC/WRS`. The first two jumpers are either closed or open, while the last jumper is an alternative jumper that prevents / enables EEPROM programming. `JP4` controls if the code should be executed directly in ROM. It overwrites `JP5`. `JP5` decides if the code execution starts after the code is copied from ROM to RAM. The nominal positions for the jumpers are: `JP4` open, `JP5` closed, `VCC` closed.

For both the SEM the following programming procedure has to be followed:

Step	Action	Result
1	Switch PSD off	Makes sure that PSD is powered off
2	Take `JP5` off	Prevents that code executes immediately after transfer from ROM to RAM. Code will stay in a testing loop
3	Switch PSD on	Power PSD on to enable EEPROM writing
4	Switch jumper `VCC` to `WRS`	Enables writing to EEPROMs
5	Run `ENGPROG.BAT`	Programs EEPROMs page after page. Press key after each page.
6	Switch jumper `WRS` to `VCC`	Disables EEPROM writing
7	Put `JP5` on	Enables code execution

The purpose of the batch file `ENGPROG.BAT` is the writing of the `256` Byte binary images to the EEPROM pages. Supposing that the splitted image files have the names `"eng.xxx"`, where `"xxx"` is a running number of the different images, the batch file starts like

```
load32          eng.b00          088
pause
load32          eng.b01          100
pause
load32          eng.b02          200
pause

...
```

where

```
load32 <file> <address>
```

is an executable that loads a binary image `<file>` into the PSD memory, starting from `<address>` (given in hexadecimal). After each image load, the `pause` command assures that a certain time elapses (the user has to press a key). This time is necessary for proper EEPROM programming.

After programming it is advised to perform a verification of the data that has been written to the EEPROM. This can be done using the executable

```
save32 <file> <address> <length>
```

where `<file>` is the name of the file to which the binary image is written, `<address>` is the starting address from which the EEPROM memory should be read, and `<length>` is the number of Bytes that should be read. Both `<address>` and `<length>` are given in hexadecimal.

Example: Suppose the code that has been written had a length of `0x4000` (hexadecimal). Then using `"save32 eng.rom 0x88 0x4000"` this code is read back from EEPROM and saved into the file `eng.rom`. This file can then be compared to the original code binary in order to check if the EEPROM write was successful.

Another possibility to look directly into the PSD memory is given by the executable

```
disp32 <address> <length>
```

The format is similar to the `save32` command, except that the information is not written to a file but displayed on the screen.

6.3. Programming the loader

The loader is the part of the PSD code that loads the program from EEPROM into RAM. It is situated at the memory address `0x000000` and has a length of `136` Bytes. Assuming that the loader resides in a binary file called "`loader.bin`", the loader code may be loaded into the PSD using the command

```
load32 loader.bin 000
```

This could become necessary when the loader code in the EEPROM becomes corrupted due to a bad EEPROM programming manipulation. Normally, the loader needs not to be replaced.

7. Housekeeping telemetry values

All telecommands sent to the PSD SEM are registered in memory and resent as answer in case of TC readback requests. Only the following parameters lead to a change in SEM functionality:

Database Identifier	TC (hex)	Parameter
	0A	Number of curves per 8 Hz cycle for OPER
	0A	Periodicity for sending one curve for OPER
	0A	Number of curves per 8 Hz cycle for CAL & DIAG
	0A	Periodicity for sending one curve for CAL & DIAG

In particular, library uploading in the SEM has no effect.

Except of the error counter (Exxx) and the last error type (Exxx) all housekeeping telemetry is kept fixed in the SEM. The following tables specify the housekeeping parameters that are expected for each parameter block (see also Annex 21.4 for the definition of the parameter fields). For Exxx and Exxx the default values expected after start-up are given in parentheses.

Database Identifier	Value (hex)	Parameter
	00	Number of multiple events from detector 0
	01	Number of single events from detector 0
	00	Number of multiple events from detector 1
	02	Number of single events from detector 1
	00	Number of multiple events from detector 2
	03	Number of single events from detector 2
	00	Number of multiple events from detector 3
	04	Number of single events from detector 3
	00	Number of multiple events from detector 4
	05	Number of single events from detector 4
	00	Number of multiple events from detector 5

06	Number of single events from detector 5
00	Number of multiple events from detector 6
07	Number of single events from detector 6
00	Number of multiple events from detector 7
08	Number of single events from detector 7
00	Number of multiple events from detector 8
09	Number of single events from detector 8
00	Number of multiple events from detector 9
10	Number of single events from detector 9
00	Number of multiple events from detector 10
11	Number of single events from detector 10
00	Number of multiple events from detector 11
12	Number of single events from detector 11
00	Number of multiple events from detector 12
13	Number of single events from detector 12
00	Number of multiple events from detector 13
14	Number of single events from detector 13
00	Number of multiple events from detector 14
15	Number of single events from detector 14
00	Number of multiple events from detector 15
16	Number of single events from detector 15
00	Number of multiple events from detector 16
17	Number of single events from detector 16
00	Number of multiple events from detector 17
18	Number of single events from detector 17
00	Number of multiple events from detector 18
19	Number of single events from detector 18
01	Average baseline for detector 0
01	Noise for detector 0
01	Average baseline for detector 1
02	Noise for detector 1
01	Average baseline for detector 2
03	Noise for detector 2
01	Average baseline for detector 3
04	Noise for detector 3
01	Average baseline for detector 4
05	Noise for detector 4

01	Average baseline for detector 5
06	Noise for detector 5
01	Average baseline for detector 6
07	Noise for detector 6
01	Average baseline for detector 7
08	Noise for detector 7
01	Average baseline for detector 8
09	Noise for detector 8
01	Average baseline for detector 9
10	Noise for detector 9
01	Average baseline for detector 10
11	Noise for detector 10
01	Average baseline for detector 11
12	Noise for detector 11
01	Average baseline for detector 12
13	Noise for detector 12
01	Average baseline for detector 13
14	Noise for detector 13
01	Average baseline for detector 14
15	Noise for detector 14
01	Average baseline for detector 15
16	Noise for detector 15
01	Average baseline for detector 16
17	Noise for detector 16
01	Average baseline for detector 17
18	Noise for detector 17
01	Average baseline for detector 18
19	Noise for detector 18
44	RAM parameter checksum verification
00	Do not care
1	Global Front End Trigger
0	Low Level Discriminator
0	Time window
0	Gain control
1	Disable / Enable detector 0
1	Disable / Enable detector 1
1	Disable / Enable detector 2

	1	Disable / Enable detector 3
	1	Disable / Enable detector 4
	1	Disable / Enable detector 5
	1	Disable / Enable detector 6
	1	Disable / Enable detector 7
	1	Disable / Enable detector 8
	1	Disable / Enable detector 9
	1	Disable / Enable detector 10
	1	Disable / Enable detector 11
	1	Disable / Enable detector 12
	1	Disable / Enable detector 13
	1	Disable / Enable detector 14
	1	Disable / Enable detector 15
	1	Disable / Enable detector 16
	1	Disable / Enable detector 17
	1	Disable / Enable detector 18
	0000	Number of thrown away events
E3824	0101	+5 V digital
E3825	0102	+5 V analog
E3826	0103	-5 V analog
E3827	0104	A/D global offset (voltage reference)
E3828	0201	DSP non memory board temperature
E3829	0202	A/D board temperature
E3830	0203	Analog MUX2 board temperature
E3831	0204	Analog MUX1 board temperature
E3832	0001	Command count
E3833	4D	Last received command code
E3834	58	Last received command identifier
E3835	0020	Last HSL identifier sent to DFEE
E3836	0001	8 Hz counter
E3837	05	Events in HSL buffer 0
E3838	05	Events in HSL buffer 1
E3839	01	Curves in HSL buffer 0
E3840	01	Channel rate for detector 0
E3841	02	Channel rate for detector 1
E3842	03	Channel rate for detector 2
E3843	04	Channel rate for detector 3

E3844	05	Channel rate for detector 4
E3845	06	Channel rate for detector 5
E3846	07	Channel rate for detector 6
E3847	08	Channel rate for detector 7
E3848	09	Channel rate for detector 8
E3849	0A	Channel rate for detector 9
E3850	0B	Channel rate for detector 10
E3851	0C	Channel rate for detector 11
E3852	0D	Channel rate for detector 12
E3853	0E	Channel rate for detector 13
E3854	0F	Channel rate for detector 14
E3855	10	Channel rate for detector 15
E3856	11	Channel rate for detector 16
E3857	12	Channel rate for detector 17
E3858	13	Channel rate for detector 18
E3859	01	Curves in HSL buffer 1
E3879	(01)	Error count since last Power-On
E3880	(20)	Last error type
E4781	0100	LLD rate for 1st interval
E4782	0300	LLD rate for 2nd interval
E4783	0500	LLD rate for 3rd interval
E4784	0700	LLD rate for 4th interval
E4785	0900	LLD rate for 5th interval
E4786	0B00	LLD rate for 6th interval
E4787	0D00	LLD rate for 7th interval
E4788	0F00	LLD rate for 8th interval
E4789	1100	LLD rate for 9th interval
E4790	1300	LLD rate for 10th interval
E4791	1500	LLD rate for 11th interval
E4792	1700	LLD rate for 12th interval
E4793	1900	LLD rate for 13th interval
E4794	1B00	LLD rate for 14th interval
E4795	1D00	LLD rate for 15th interval
E4796	1F00	LLD rate for 16th interval
E4797	2100	LLD rate for 17th interval
E4798	2300	LLD rate for 18th interval
E4799	2500	LLD rate for 19th interval

E4800	2700	LLD rate for 20th interval
E4801	2900	LLD rate for 21st interval
E4802	2B00	LLD rate for 22nd interval
E4803	2D00	LLD rate for 23rd interval
E4804	2F00	LLD rate for 24th interval
E4805	3100	LLD rate for 25th interval
E4806	3300	LLD rate for 26th interval
E4807	3500	LLD rate for 27th interval
E4808	3700	LLD rate for 28th interval
E4809	3900	LLD rate for 29th interval
E4810	3B00	LLD rate for 30th interval
E4811	3D00	LLD rate for 31st interval
E4812	3F00	LLD rate for 32nd interval
E4821	0200	ULD rate for 1st interval
E4822	0400	ULD rate for 2nd interval
E4823	0600	ULD rate for 3rd interval
E4824	0800	ULD rate for 4th interval
E4825	0A00	ULD rate for 5th interval
E4826	0C00	ULD rate for 6th interval
E4827	0E00	ULD rate for 7th interval
E4828	1000	ULD rate for 8th interval
E4829	1200	ULD rate for 9th interval
E4830	1400	ULD rate for 10th interval
E4831	1600	ULD rate for 11th interval
E4832	1800	ULD rate for 12th interval
E4833	1A00	ULD rate for 13th interval
E4834	1C00	ULD rate for 14th interval
E4835	1E00	ULD rate for 15th interval
E4836	2000	ULD rate for 16th interval
E4837	2200	ULD rate for 17th interval
E4838	2400	ULD rate for 18th interval
E4839	2600	ULD rate for 19th interval
E4840	2800	ULD rate for 20th interval
E4841	2A00	ULD rate for 21st interval
E4842	2C00	ULD rate for 22nd interval
E4843	2E00	ULD rate for 23rd interval
E4844	3000	ULD rate for 24th interval

E4845	3200	ULD rate for 25th interval
E4846	3400	ULD rate for 26th interval
E4847	3600	ULD rate for 27th interval
E4848	3800	ULD rate for 28th interval
E4849	3A00	ULD rate for 29th interval
E4850	3C00	ULD rate for 30th interval
E4851	3E00	ULD rate for 31st interval
E4852	4000	ULD rate for 32nd interval