HERSCHEL/PACS

SPU HIGH LEVEL SOFTWARE
User Manual

Document Ref.: PACS-TW-HM-002

Issue: 13.96

Prepared by: **Roland Ottensamer** and **Christian Reimers** Date: **18 Feb 2009**

Distribution Record

Issue / Revision	Draft 1	Issue 1.0	Issue 1.1	Issue 1.2	Issue 1.3	Issue 1.4	Issue 2.0
Distribution Date	03.04.2002	07.08.2002	11.09.2002	23.09.2002	26.09.2002	26.09.2002	29.09.2002
Issue / Revision	Issue 2.1	Issue 2.2	Issue 2.3	Issue 2.4	Issue 3.0	Issue 3.1	Issue 3.2
Distribution Date	02.10.2002	13.11.02	29.11.02	03.12.02	21.02.03	13.03.03	11.04.03
Issue / Revision	3.3	3.4	3.5	4.0	5.0	5.6	5.7
Distribution Date	09.05.2003	14.05.2003	14.05.2003	26.05.2003	03.07.2003	09.07.2003	30.07.2003
Issue / Revision	5.8	6.0	6.1	8.0	8.6	9.0	10.0
Distribution Date	31.07.2003	16.09.2003	26.09.2003	27.02.2004	30.03.2004	11.05.2004	24.05.2004
Issue / Revision	10.1	11.0	11.1	11.2	11.4	11.5	11.7
Distribution Date	15.06.2004	19.07.2004	23.07.2004	31.08.2004	08.06.2005	22.06.2005	22.08.2005
Issue / Revision	12.0	12.1	12.2	12.8	13.2	13.5	13.8
Distribution Date	14.12.2005	23.01.2006	28.2.2006	14.8.2006	07.03.2007	3.4.2007	31.5.2007
Issue / Revision	13.8a	13.9	13.95	13.95a	13.96		
Distribution Date	20.06.2007	18.06.2008	3.02.09	5.02.09	18.02.09		

INTERNAL			**EXTERNAL**		
Department	**Name**	**Qty**	**Company**	**Name**	**Qty**
TUVIE/PRIP	F. Kerschbaum	1	MPE	H. Feuchtgruber, O. Bauer, E. Wieprecht	1
UVIE/ASTRO	C. Reimers	1			
			MPE	PACS Project Office (pacs@mpe.mpg.de) PACS Warm Electronic (pacs-we@ster.kuleuven.ac.be)	1
			Electronic Archives at Leuven http://pacs.ster.kuleuven.ac.be		1

Document Change Record

Document Title **HERSCHEL – PACS** **SPU HLSW to DPU Interface Description**		
Issue	**Date**	**Reason for Change**
Draft 1	02/04/2002	Initial issue
Issue 1.0	07/08/2002	Software Update before AVM Acceptance
Issue 1.1	10/09/2002	Including detailed installation guide
Issue 1.2	23/09/2002	- Updating the Installation Guide - Including the information where the RCX field may be used respective to Compression Mode
Issue 1.3	26/09/2002	- Updating the Initialisation of DXS Tables
Issue 1.4	26/09/2002	- Including the Software update respective to the new DPU SPU protocol (AD002 Issue 4.0). - Update of Running Confidence Check Section respective to the HLSW Version 1.2
Issue 2.0	29/09/2002	- Update of the Installation guide (Build TCs to Upload the Software via the Nominal Route: CDMS-DPU) - Update of Running Confidence Check Section respective to the HLSW Version 2.0 - Including the description of the DEC/MEC Header ‘s Label and Validity fields - Including the expected TM Rate in all compression modes
Issue 2.1	02/10/2002	- Correct WSDP table length (0x0A) to 0x18 words - Update Parameter 5 (length of pmco segment) in command to Load HLSW from EEPROM to RAM - Update picture of HLSW Memory Map in SPU EEPROM (end adress of pmco segment) - Including detailed description in WSDP table fields SPUID and CMM. - Including a summary of supported DPU commands with command structure and parameters (Table 7).
Issue 2.2	13/11/2002	- Update of Running Confidence Check Section respective to the HLSW Version 2.1
Issue 2.3	29/11/2002	- Update of Running Confidence Check Section respective to the HLSW Version 2.2 - Include in the HK Packets the received Number of DEC/MEC Packets
Issue 2.4	03/12/2002	- Update of Running Confidence Check Section respective to the HLSW Version 2.3 - Include in the HK Packets the CPU Workload instead of the Number of Maintained Ramps

3.0	21/02/2003	- Update of Running Confidence Check Section respective to the HLSW Version 3.0 - Include information about SPU simulated data
3.1	13/03/2003	- Include the threshold for chopper and grating positions deviation in the detector constants tables - Update of Running Confidence Check Section
3.2	11/04/2003	- Update of Running Confidence Check Section respective to the HLSW Version 3.1 - Include the command Copy Data from RAM to EEPROM - Update the HK and TC naming convention according to the MIB
3.3	09/05/2003	- Update of Running Confidence Check Section respective to the HLSW Version 3.2 - Include the command CONNECT_DMC - Update the HK Tables according to the DPU-SPU HLSW ICD issue 4.2 - Update the RCX parameter definition in the RAW_CHAN_TRANS_MODE command
3.4	14/05/2003	- Update of Running Confidence Check Section respective to the HLSW Version 3.4
3.5	14/05/2003	- Update of Running Confidence Check Section respective to the HLSW Version 3.5 - Include the SPU HLSW requirements to send the CONNECT_DMC Command - Change the Size of the HK packet according to the Updated DPU-SPU HLSW ICD
4.0	26/05/2003	- Update of Running Confidence Check Section respective to the HLSW Version 4.0 - Include the SMCS Chip Handling Description (Section 4.4) - Add the Length of the data to be copied from RAM to EEPROM under Paragraph CP_DATA_RAM_EEPROM - Modify DMC_Link_Status information (0x00: Connection is OFF and 0xFF: Connection is ON)
5.0	03/07/2003	- Update of Running Confidence Check Section respective to the HLSW Version 5.0 - Add the Compressed Raw Channel Size in Decompressed Entity Header instead of the spare field (Section 5.5) - Update the Description of RAW_CHAN_TRAN_MODE in Section 5.1.1. to declare that data from the selected channels are compression lossless. - Correct the Parameter Ids is Table 10 to be compatible with those in Table 9. - Add Clarifications in Section 5.1.1. about the use of the WRT_DXSx commands
5.6	07/07/2003	- Update of Running Confidence Check Section respective to the HLSW Version 5.6 - Updated description (detailed) of the compressed data (Section 4.3.3) - Include the sub-ramp length (for ramp fitting) in the Detector Constant Table for Spectroscopy (Section 5.1.1.)
5.7	29/07/2003	- Update of Running Confidence Check Section respective to the HLSW Version 5.7

		- Update of the description of sub-ramp length (for ramp fitting) in the Detector Constant Table for Spectroscopy (Section 5.1.1.) - Table 16 is updated. Include the error code "invalid parameter" for the Write command
5.8	31/07/2003	- Update of Running Confidence Check Section respective to the HLSW Version 5.8
6.0	16/09/2003	- Update of Running Confidence Check Section respective to the HLSW Version 6.0 - Warning is added in Table 10 - Include Section 5.7 with Typical SW CPU Workload and Data Rate - Update the naming conventions in Section 5.2 according to the DEC/MEC to SPU ICD issue 3.4
6.1	26/09/2003	- Update the data rates tables in Section 5.5.2 respective to the SW specifications
7.0	12/07/2003	- BOLC Status Useful bytes are 2 bytes
8.0	27/02/2004	- Update of Running Confidence Check Section respective to the HLSW Version 8.0 - Include a parameter for the Half Compression Mode in Photometry (20Hz readout rate) - Modify the DIFS field in Compressed entity Header by REAL (reduction algorithm) used on-board. Includes the possible codes for this parameter - Include two parameters for the Write command in spectroscopy (Reduction algorithm to use, the use of glitch or not) - Update the CPU workload respective to the new HLSW run on 6-Pack data - Include typical CPU workload for the run of HLSW with simulated data
8.6	29/03/2004	- Update of Running Confidence Check Section respective to the HLSW Version 8.6 - Noise resampling algorithm is implemented in the peak-up module. Its description is under the Peak-up command Section 5.1.1. - Update the HK Table parameters Section 5.4.1. Put default values for HK parameters in SPU idle mode. Deleting the CR and HBC parameters and replacing them respectively by REAL, MEM_STATUS_CNTS and SPARE. Update the definition of CI in the HK table - Include the recommendation of How to monitor memory errors in HK Section 5.4.2. - Update the compression parameters list in Table 14 - Updated memory map in Appendixes A and B
9.0	11/05/2004	- Update of Running Confidence Check Section respective to the HLSW Version 9.0 - Update of the memory map in the Appendices. - Update the Section 5.4.2. such that the actual SPU SW also checks for memory errors in EEPROM. - Include notes about the TM buffer size in Section 5.5.
10.0	24/05/2004	- Update of Running Confidence Check Section respective to the HLSW Version 10.0

		- Include one parameter for the command "Write detectors constant in photometry" (The use of glitch detection is optional) - The parameter "Validity" in the DEC/MEC Header in photometry is not checked by this version of the software for CPU workload reason See Section 5.2.
10.1	01/06/2004	- Update of Running Confidence Check Section respective to the HLSW Version 10.1 - Update Table 20 for the Reduction algorithms codes. - Update the note about the TM buffer size in Section 5.5. - Update of the memory map in the Appendices. - Update of the memory addresses for the memory scrubbing reporting in Section 5.4.2.
11.0	17/07/2004	- Update of Running Confidence Check Section respective to the HLSW Version 11.0 - Include Table 19 for DMC header errors description
11.1	23/07/2004	- Update of Running Confidence Check Section respective to the HLSW Version 11.1
11.2	31/08/2004	- Update of Running Confidence Check Section respective to the HLSW Version 11.2
11.4	08/06/2005	- Update of Running Confidence Check Section respective to the HLSW Version 11.4 - In Table 9, the PAR ID definition is updated. - In Table 17, the SPU_RCX definition is updated.
11.5	08/06/2005	- Update of Running Confidence Check Section respective to the HLSW Version 11.5
11.7	22/08/2005	- Update of Running Confidence Check Section respective to the HLSW Version 11.7 - Update of Table 4. Background cancelling modules have been removed - Update of Table 5. Background cancelling command has been removed - Update of Table 7. Background cancelling command has been removed - Background cancelling command description has been removed from Section 5.1.1 - Add a warning in Section 5.2.1 about the maximum allowed number of selected detectors in lossless compression mode. - Add a warning in Section 5.2.1 about the maximum allowed number of samples per subramp in the spectroscopy compression mode for 4s reset. - Add a note in Section 5.2.1 about the double compression mode in spectroscopy. - Update of Table 22. Data rate in double compression is equivalent to the one in nominal mode.
12.0	13/12/2005	- Update of Running Confidence Check Section respective to the HLSW Version 12.0 - Update of the list of SW module in Section 4.1 - Update of the write parameters in spectroscopy in Table 10 *Update of the write parameters in photometry in* - Table 11 - Update of the write parameters in Table 12 - Updates in Table 17:

		- PIX definition is updated - SPARE_2 and SPARE_3 have been replaced by LLC_ERROR and PAR_MONITOR respectively. - Section 5.8 is included
12.1	23/01/2006	- Update of Running Confidence Check Section respective to the HLSW Version 12.1
12.2	28/02/2006	- Update of Running Confidence Check Section respective to the HLSW Version 12.2
12.8	14/08/2006	- LLSW driver version number given in section 1.2 - Tables 3 and 4 (checksums) updated - List of SW modules (Table 5) modified: added “bol_ex.c”, “T_S_fm.c”, “rolib_a.s” - Table 11 (Write Detector Constants in Spectroscopy) modified - Table 12 (Write Detector Constants in Spectroscopy) modified - Table 18 (List of HK entries) updated
13.2	07/03/2007	- Tables 3, 4, 5, 11, 12, 13, 19, 22, 28, 30 modified - Text in subsection 5.2.1 modified - Subsection 5.7.2 rewritten
13.5	03/04/2007	- Subsection 5.7.1. removed. Section hierarchy adapted accordingly - Tables 3, 4, 9, 11, 12, 13, 28 modified - Appendix B.1 corrected
13.8	31/05/2007	- Tables 3, 11, 17 and 25 updated
13.8a	20/06/2007	- Description of RAW_CHAN_TRAN_MODE on page 40 revised - Hint on units added on page 51
13.9	18/06/2008	- Tables 3, 10, 11, 14 and 25 updated - Appendix E added - Appendix F added
13.95	03/02/2009	- Tables 3, 10, 11, 20 updated - Appendix F updated
13.95a	05/02/2009	- Hint added to the description on page 40 - Appendix F.1 corrected
13.96	18/02/2009	- Tables 3 and 10 updated

Contents

List of Figures

List of Tables

1. Introduction

This document is the Software User Manual of the Signal Processing Unit (SPU) High Level Software (HLSW). It refers to the software, which constitutes the Application Software (ASW) for data compression/reduction of the PACS instrument.

1.1. Intended Readership

The natural customer of this SW User Manual will be:

- The software engineers, who will follow the unrolling of the satellite lifecycle, performing tests, maintenance of the software
- The ICC personnel responsible of the command, providing the command procedures related to the HLSW parameters to be performed
- The other instrument subsystem developers related to the SPU HLSW

1.2. Applicability Statement and Configuration Constraints

This SW User Manual is applicable to the SPU HLSW, version 13.9 dated of 13.6.2008.

The target hardware required to run the software has the following configuration (see RD003):

- TSC21020F IEEE 32/40 bits Floating Point Digital Signal Processing at up to 18 MHz, 0 wait states for RAM
- 32k x 48 bits start-up PROM
- 256k x 48 bits EEPROM program bank (EDAC protected)
- 512k x 48 bits Program RAM (EDAC protected)
- 512k x 32 bits Data RAM plus an 512k x 40 bits Expansion RAM (both banks EDAC protected)
- 3 bi-directional IEEE-1355 controlled by SMCS332 capable of 100 Mbps data rates

The software recommended to compile the software is:

- Window NT 4.0
- Virtuoso real time OS (version 4.2, revision 3) with library
- LLSW drivers from CRISA (version 1.4)
- ADSP 21k library
- G21k Compiler and ld21k Linker (version 3.3.)

1.3. Purpose

This document describes how to use the HLSW on the target machine. The main functions of the software are:

- Acknowledge and Execution of DPU commands
- Compression/Reduction of Science Data from DEC/MEC due to the compression parameters sent via DEC/MEC Header to SPU or tables stored on board.
- Generate HK, which reports the status and information of the SPU HLSW.

The purpose of the SPU HLSW is the data compression/reduction within the PACS instrument on board the Herschel satellite.

1.4. Structure of the document

The User Manual of the SPU HLSW is structured as follows:

- In Section 1, an introduction on the use of this document is presented.
- In Section 2 a short description of the SPU HLSW is given.
- In Section 3, the installation procedure is presented.
- Section 4 contains the instruction for each task.
- Section 5 summarizes the SPU HLSW environment.

1.5. Related Documents

1.5.1. Applicable Documents

AD001	PACS-TW-GS-001	SPU High Level Software Specification Document
AD002	PACS-TW-ID-001	SPU HLSW to DPU Interface Description
AD003	PACS-TW-SR-001	HERSCHEL/PACS SPU HLSW User Requirement Document

1.5.2. Reference Documents

RD001	BSSC(96)2	Guide to applying the ESA software engineering standards to small Software projects
RD002	PACS-CL-ID-004	Interface Control Document DEC/MEC – SPU
RD003	FPL-IC-1214-01-CRS	PACS SPU HW-SW Interface Control Document
RD004	IFSI/OBS/PL/2000-001	DPU/ICU On board Software Product Assurance Plan
RD005	DIPSAPII-DAS-31-06 issue 2	SMCS332 user manual
RD006	VUG42R300BX	Wind River Virtuoso user guide (version 4.2, revision 3)
RD007	PACS-IC-TN-001	PACS and LFI SPU_SUSW and DPU_ASW Protocol
RD008	PACS-IC-PL-001	PACS SPU Limited Functional Tests
RD009	SCI-PT-ICD-7527	Packet Structure – Interface Control Document (PS-ICD)
RD010	FPL-IC-1214-04-CRS	SPU LLSW Drivers SW Interface Control Document
RD011	FPL-SRD-1214-01-CRS	SPU Start-up and Low Level Software Drivers
RD012	PACS-TW-LI-001	SPU HLSW Configuration Item Data List v1.2 (__.11.2006)

1.6. Conventions

To highlight important messages following symbols are used in this user manual:

☞ **Note**: point out something important or useful to know.

Caution: tell you about commands or procedures that may have unwanted or undesirable side effects.

Warning: tell you about commands or procedures that could be dangerous. This means: It is advisable to follow this warnings, to do not interrupt continuous operation of commanding process or compression SW, but is has no „dangerous“ impact to the instruments health.

In the following Table 1 stylistic and command syntax conventions are given.

DPU Commands	
LOAD	Load
DUMP	Dump
CHECK	Check
CP_DATA_RAM_EEPROM	Copy Data from RAM to EEPROM
RESET	Warm Reset
RAW_CHAN_TRAN_MODE	Raw Channel Selection
STOP_REDUCT_COMPR	Stop
START_REDUCT_COMPR	Start
ACT_TEST_PHOT	SPU Test Mode for Photometry
ACT_TEST_SPEC	SPU Test Mode for Spectroscopy
DMC_CONNECT	DEC/MEC Connect
WRT_DXSx	Write Detectors Selection Table (x = 1…7)
WRT_DET_CST_SPEC	Write Detectors Constants in Spectroscopy
WRT_DET_CST_PHOT	Write Detectors Constants in Photometry
WRT_SIM_DATA	Write Simulated Data Parameters
DEC/MEC header	
SPUID	SPU ID
OBSID	Observation Identification
BBID	Building Block Identification
LBL	Label
TMP	Timing Parameters
ROSS	Readout Specifications in Spectroscopy
ROSP	Readout Specifications in Photometry
BSID	Bolometer Setup Identification
VLD	Validity
CPR	Chopper Position Readback
WPR	Wheel Position Readback
GPR	Grating Position Readback
BOLC_ST	BOLC STatus
CRDC	Current ReaDout Count
RRR	Readouts in Ramp Readback
CRCRMP	Current Readout Count in a RaMP
CRECR	CRE Control Readback
CRDCCP	Current ReaDout Count in Chopper Position
DBID	Data Block ID
Housekeeping (HK)	
HK_HEADER	HK Header
OBSID	Observation Identification
PIX	Packet Index
CI	Counter Increments
REAL	Reduction Algorithm
SATUR_FLAG	Saturation Flag
SAMP_CORR	Glitch Counter information

MAINT_RAMPS	Number of Maintained Ramps/Averages
CPU_WORKLOAD	CPU Workload
DMC_LINK_STATUS	DEC/MEC Link Status
INTEG_RAMPS	Number of Integrated Ramps/Averages
VID	Version ID
RCX	Raw Channel indeX
DMC_ERROR	DEC/MEC Header Error
MEM_STATUS_CNTS	Memory Status Counters
Command Acknowledgement	
PACK ID	Positive Acknowledgement Identification
NACK ID	Negative Acknowledgement Identification
Compressed Science Data	
CEH	Compressed Entity Header
CDH	Compressed DEC/MEC Header
CSD	Compressed Science Data
PIX	Packet Index
DECID	DEcompression Code IDentification
REAL	Reduction ALgorithm
RCX	Raw Channel indeX
CMM	Compression Mode
DXS ID	Detectors Selection table IDentification
CRCS	Compressed Raw Channel data Size
CDHS	Compressed DEC/MEC Header Size
SCIS	compressed SCIence data Size
Other	
ADU	Analog/Digital Unit

Table 1. Stylistic and Command Syntax Conventions

1.7. Problem reporting instructions

See section 6 of the RD004 (Issue 1) document for the problem reporting instructions.

2. Overview

The HLSW of the SPU is responsible for data processing (reduction/compression), control of the DEC/MEC and DPU communication link and the execution and acknowledgement of several DPU commands. The HLSW consists of a set of program sequences for several tasks. The HLSW is split into three main parts: the ASW, the watch process and the communication interface.

- The ASW performs data reduction/compression according to the DEC/MEC header and/or tables stored on board. The compression mode to be achieved is defined by the supervisor program, which is set at the top of the ASW.

- The watch process represents a program which link to the DPU link, waits for commands, decodes them and interrupts the ASW for the acknowledgment of the DPU command as described in AD002. Each DPU command has to be acknowledged.

- The communication interface uses the LLSW drivers to read and write data on the IEEE 1355 link. It has the responsibility to receive data packets from DEC/MEC and DPU and to send data packets to DPU.

The SPU CPU module is based on the ADSP 21020 processor type. The communication of DEC/MEC to SPU and DPU to/from SPU is provided by IEEE-1355 communication links (see RD005), implemented by the SMCS332 chip (see RD005). The operating system of the HLSW is the Virtuoso real time OS (RD006).

The inputs to the HLSW should be raw data with a header (as described in RD002) on the DEC/MEC link and/or commands (as described in AD002) on the DPU link.

3. Installation Guide

3.1. Introduction

✋ Before installing the SPU HLSW:

- Read Section 3.1 (this section) carefully.
- Section 3.2: HLSW Upgrade covers the actual installing process for the SPU HLSW in detail. The section Running Confidence Check covers the confidence test of the HLSW configuration.
- Section 3.3: Running SPU HLSW covers the actual procedure to start-up and initialise the Software.

This guide explains how to install the SPU HLSW. It describes the Install shield procedure, and shows you to turn a number of confidence tests to ensure the installation is successful.

☞ The HLSW is an identical product for both SPU boards (SWL and LWL). Therefore, the SW installation procedure is identical.

3.1.1. Supported Boards

This installation concerns the SPU-AVM with the specifications listed in the document RD003. The processor module is a Digital Signal Processing CPU on a single board based on the DSP TSC21020F, Radiation Tolerant version of ADSP21020 from Analog Devices.

☞ Please remember to validate the correct working of the SPU board with the vendor's confidence test (RD008).

3.1.2. Configuration

The IEEE 1355 Link of the SPU HLSW is configured as a slave to DPU SW and DEC/MEC SW referring to the IEEE 1355 Link protocol (RD005). In Figure 1, you can see the SPU HLSW IEEE 1355 Link Configuration.

When you start the SPU HLSW the DPU and DEC/MEC SWs must be already running.

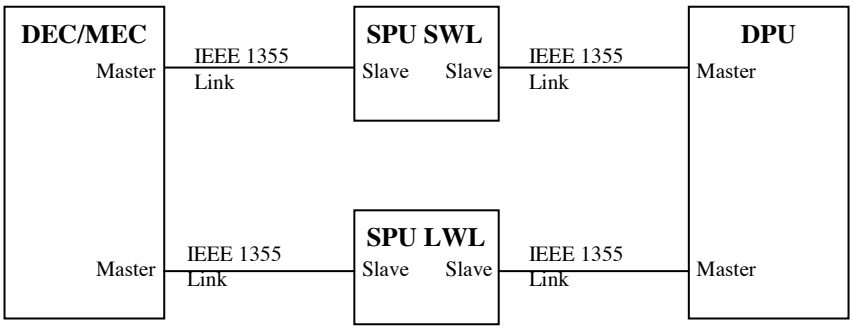


Figure 1. SPU HLSW IEEE 1355 Link Configuration

3.1.3. Contact

If you encounter any problems using SPU HLSW or have any questions about it, we will be happy to help you. You can reach us using following E-mail address:

spuhlsw@prip.tuwien.ac.at

3.1.4. Root Folder

The default installation root folder of the SPU HLSW Installation SW Package is **`C:\Spu_hlsw`** with the folder structure presented in Table 2.

3.2. HLSW Upgrade

3.2.1. Updating from a Previous Version

If you have already installed an older version of the HLSW in SPU EEPROM and want to update it, follow these steps:

- Uninstall the previous version of the HLSW (see next section).
- Follow the steps for installation of the HLSW in SPU EEPROM.

☞ It is recommended to update the HLSW in both SPU boards (SWL and LWL) if required.

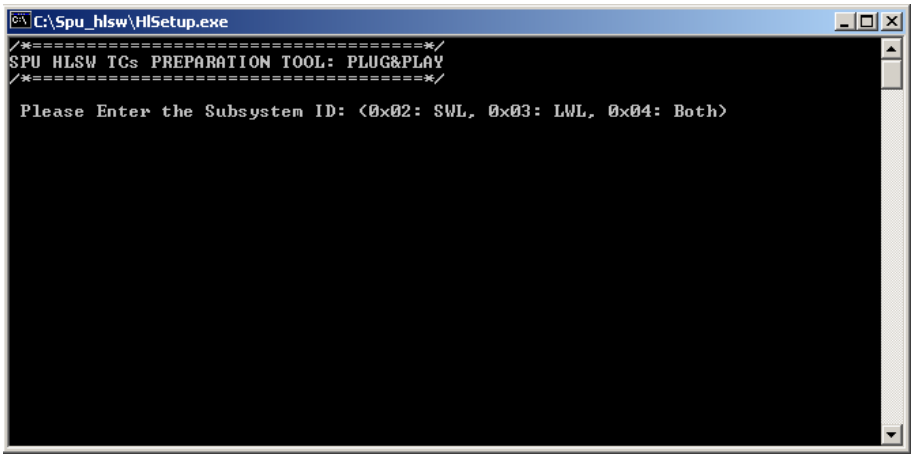
3.2.2. Uninstalling old Version of SPU HLSW

Before to uninstall the HLSW on the SPU you have to build TC packets (according to RD009) which contains data with value 0.

Using our tools you should follow the steps described below:

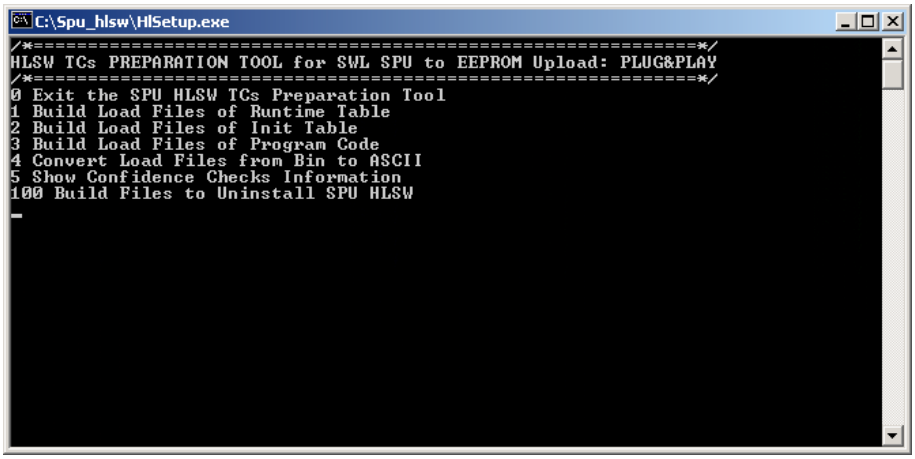
Preparation of TC packets

- Open `C:\Spu_hlsw\HlSetup.exe` on the PC



- Type the subsystem ID (2 for SWL SPU, 3 for LWL SPU or 4 for both SPUs)
- Type the Memory Block (1 for SPU RAM or 3 for SPU EEPROM)

For example, the following picture shows the menu displayed when you typed for the subsystem ID = 2 (SWL SPU) and Memory Block = 3 (SPU EEPROM):



- Type 100 to build the binary Files to uninstall the SPU HLSW.

Now there should be binary Files named `SCOSLD.xxx` (`xxx` stands for a counter which starts by `000`) on the PC.

- Type 4 to convert the binary Files into ASCII files.

Now there should be ASCII files named `Loadxxxx.SPU` (`xxxx` stands for a counter which starts by `0001`) on the PC

- Type 0 to exit `C:\Spu_hlsw\HlSetup.exe`

HLSW Uninstallation

- To uninstall the HLSW on the SPU target machine follow the nominal route to upload the generated TC packets with SCOS2000 via CDMS and DPU into the SPU.

3.2.3. Installing the SPU HLSW

3.2.3.1. HW Setup

To install the SPU HLSW the HW Configuration must be:

- One PC to generate the TC packets which are needed to upload the SPU HLSW into the SPU target machine via the nominal route: SCOS2000 - CDMS simulator – DPU – SPU

3.2.3.2. SW Setup

The following software should be available on the PC:

- Windows NT 4.0
- MS Visual C++

☞ The installation has been validated using Windows NT. The use of other Windows operating system is not recommended.

3.2.3.3. Tools

To install the needed SPU HLSW tools for installation follow the steps below:

1. Insert the SPU HLSW CD into the CDROM drive of the PC
2. Copy the file `<CDdrive>:\RESOURCE\SOFTWARE\SPU_Pack.zip` into the PC under folder `C:\`
3. Unzip the file `SPU_Pack.zip` . Table 2 shows the list of needed files on the PC after unzip.

Folder …	Content …
C:\Spu_hlsw\	HlSetup.exe
	Hls-init.spu
	Hls-pmco.spu
	Hls-rth.spu
	splitHlsw.bat
	SPU_Hlsw.exe
	spuhlsw.ach

Table 2. Folder Structure of the SPU HLSW Installation SW package

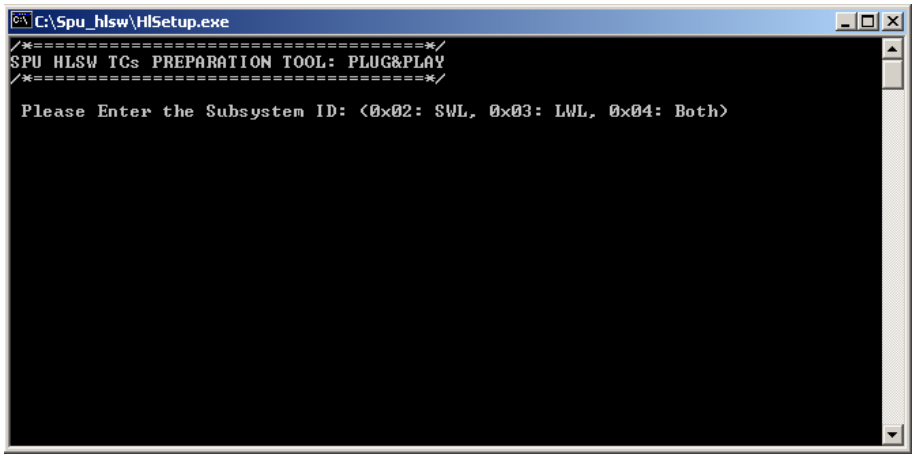
3.2.3.4. Installation Procedure

Before to install the HLSW on the SPU you have to build TC packets (according to RD009) out of the three HLSW image files `Hls-init.spu`, `Hls-pmco.spu` and `Hls-rth.spu`.

Using our tools you should follow the steps described below:

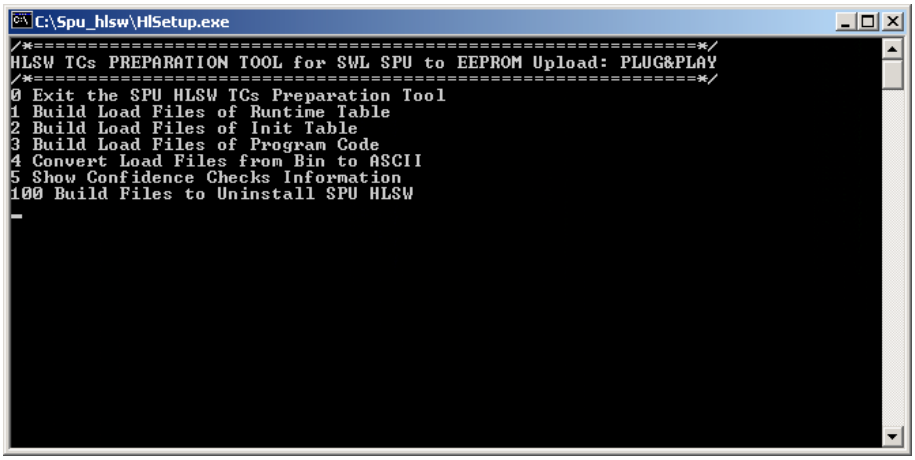
Preparation of TC packets

- Open `C:\Spu_hlsw\HlSetup.exe` on the PC



- Type the subsystem ID (2 for SWL SPU, 3 for LWL SPU or 4 for both SPUs)
- Type the Memory Block (1 for SPU RAM or 3 for SPU EEPROM)

For example, the following picture shows the menu displayed when you typed for the subsystem ID = 2 (SWL SPU) and Memory Block = 3 (SPU EEPROM):



- Type 1 to build the binary SCOS Load Files of the Runtime Table.
- Type 2 to build the binary SCOS Load Files of the Init Table.
- Type 3 to build the binary SCOS Load Files of the Program Code.

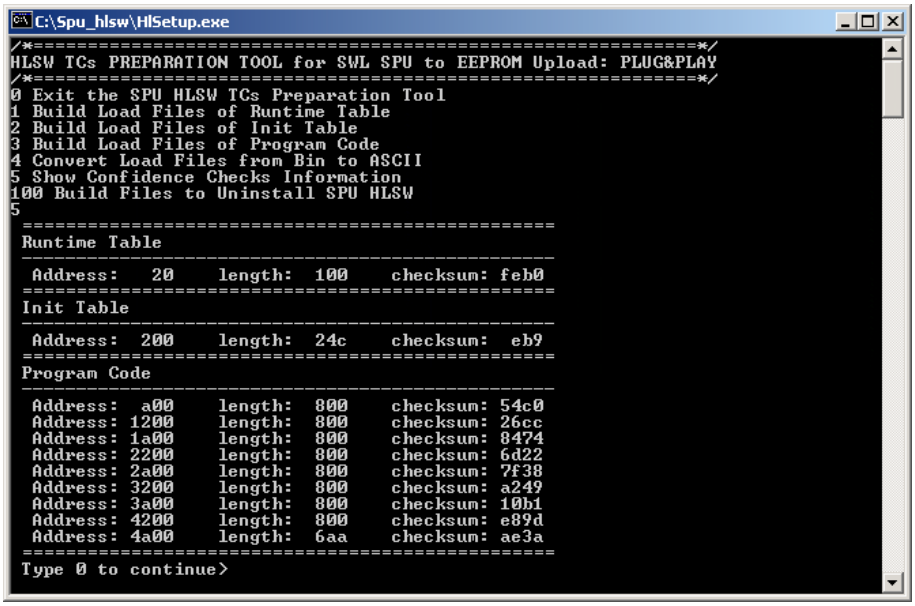
Now there should be binary Files named `SCOSLD.xxx` (`xxx` stands for a counter which starts by `000`) on the PC.

- Type 4 to convert the binary SCOS Load Files into ASCII files.

Now there should be ASCII files named `Loadxxxx.SPU` (`xxxx` stands for a counter which starts by `0001`) on the PC

- Type 5 to display the confidence check information for the Runtime Table, the Init Table and the Program Code.

For example, the following display shows the confidence check information (start address, length and checksum of the program segments)



- Type 0 to exit `C:\Spu_hlsw\HlSetup.exe`

HLSW Installation

- To install the HLSW on the SPU target machine follow the nominal route to upload the previous generated TC packets with SCOS2000 via CDMS and DPU into the SPU.

3.2.4. Running Confidence Check

These confidence checks should be used to verify that the SPU HLSW is installed correctly. Follow the steps described below:

- Switch on the SPU
- Wait 15 seconds

SPU HLSW Confidence Check for the Loaded HLSW in EEPROM

- Check runtime table, init table and program code segment in SPU EEPROM. The check of the program code segment is performed in small sub-segments. The parameters for these checks are listed below:
  - Address1 (0x4300 for Blue, 0x6300 for Red and 0x8300 for Both)
  - Adress2 (= Address taken from the table)
  - Length (see table)

- Compare the calculated checksum displayed on screen with the expected checksum (see table below)

```
Runtime Table
-------------------------------------------------
 Address:  100    length:  100    checksum: feb0
=================================================
Init Table
-------------------------------------------------
 Address:  300    length:  224    checksum: ddcd
=================================================
Program Code
-------------------------------------------------
 Address:  a00    length:  800    checksum: f921
 Address: 1200    length:  800    checksum: 6b58
 Address: 1a00    length:  800    checksum: 19d7
 Address: 2200    length:  800    checksum: a26b
 Address: 2a00    length:  800    checksum: 21f8
 Address: 3200    length:  800    checksum: 3ff2
 Address: 3a00    length:  800    checksum: 4118
 Address: 4200    length:  800    checksum: 6fb8
 Address: 4a00    length:  800    checksum:  92f
 Address: 5200    length:  800    checksum: 5d08
 Address: 5a00    length:  800    checksum: cacf
 Address: 6200    length:  800    checksum: cbfd
 Address: 6a00    length:  800    checksum: fdee
 Address: 7200    length:  800    checksum: 3b4f
 Address: 7a00    length:  800    checksum: ec6b
 Address: 8200    length:  39f    checksum: 2ab0
=================================================
```

Table 3. Confidence Check for the Loaded HLSW in EEPROM

Finish

- Switch off the SPU (HLSW installation is checked)

3.3. Running SPU HLSW

3.3.1. Starting the SPU HLSW

Running the SPU HLSW should be performed through the DPU according to the SPU SUSW to DPU communication protocol (RD007). Following steps should be done:

Step 1: **Load HLSW from EEPROM to RAM**
These commands should be sent to the SPU SUSW

Command 1: startfunc command to copy runtime header segment from EEPROM to PRAM

Command ID (2 Bytes):	`0x04`
Spare (2 Bytes):	`0x00`
Activity ID (2 Bytes):	`0x65` (Load SPU_ASW from EEPROM)
SID (2 Bytes):	`0x05`
Parameter1 (4 Bytes):	`0x03` (Memory ID for EEPROM)
Parameter2 (4 Bytes):	`0x100` (Start address of SPU_ASW in EEPROM)
Parameter3 (4 Bytes):	`0x01` (Memory ID for PRAM)
Parameter4 (4 Bytes):	`0x100` (Start address in target RAM)
Parameter5 (4 Bytes):	`0x1E0` (Length to transfer to target Mem)

Command 2: startfunc command to copy initialisation segment from EEPROM to PRAM

Command ID (2 Bytes):	`0x04`
Spare (2 Bytes):	`0x00`
Activity ID (2 Bytes):	`0x65` (Load SPU_ASW from EEPROM)
SID (2 Bytes):	`0x05`
Parameter1 (4 Bytes):	`0x03` (Memory ID for EEPROM)
Parameter2 (4 Bytes):	`0x300` (Start address of SPU_ASW in EEPROM)
Parameter3 (4 Bytes):	`0x01` (Memory ID for PRAM)
Parameter4 (4 Bytes):	`0x300` (Start address in target RAM)
Parameter5 (4 Bytes):	`0x700` (Length to transfer to target Mem)

Command 3: startfunc command to copy program memory code from EEPROM to PRAM

Command ID (2 Bytes):	`0x04`
Spare (2 Bytes):	`0x00`
Activity ID (2 Bytes):	`0x65` (Load SPU_ASW from EEPROM)
SID (2 Bytes):	`0x05`
Parameter1 (4 Bytes):	`0x03` (Memory ID for EEPROM)
Parameter2 (4 Bytes):	`0xA00` (Start address of SPU_ASW in EEPROM)
Parameter3 (4 Bytes):	`0x01` (Memory ID for PRAM)
Parameter4 (4 Bytes):	`0xA00` (Start address in target RAM)
Parameter5 (4 Bytes):	`0xA600` (Length to transfer to target Mem.)

Step 2: **SPU HLSW Confidence Check for the Copied HLSW into RAM**

- Check runtime table, init table and program code segment in SPU RAM.
  - Address1 (0x4100 for Blue, 0x6100 for Red and 0x8100 for Both)

- Compare the calculated checksum displayed on screen with the expected checksum (see Table 3)

Step 3: **Give the control to SPU HLSW**

One command should be sent to the SPU SUSW

Command ID (2 Bytes):	`0x04`
Spare (2 Bytes):	`0x00`
Activity ID (2 Bytes):	`0x66` (Give the control to SPU_ASW)
SID (2 Bytes):	`0x02`
Parameter1 (4 Bytes):	`0x01` (Memory ID for PRAM)
Parameter2 (4 Bytes):	`0xA02` (Start address in RAM of the SPU_ASW)

☞ The SPU starts sending HK after establishing connection with DPU.

3.3.2. Initialisation of the SPU HLSW

The HLSW has default DXS tables with all detectors selected (512 detectors in photometry per sub-array and 450 detectors in spectroscopy per array) and the default DXS ID is 1. The detector constant tables in spectroscopy and photometry, and the simulated data tables has initial parameters value as described in Section 5.1.1 (WRT_DET_CST_SPEC, WRT_DET_CST_PHOT and WRT_SIM_DATA).

4. Instruction

In this section a short description of the SPU HLSW input and output specifications are given. The list of SW module is given in the next section.

4.1. List of SW Modules

Both SPU boards (LWL and SWL) contain the same SW product. Hereafter the HLSW modules for one SPU sub-unit are listed:

SW Component	Files
Header Files	"bitmodel.h", "genspu.h", "globals.h", "iface.h", "macro.h", "pacscod.h", "qsmodel.h", "rangecod.h", "spuasw.h", "spu_io.h", "spulib.h"
Low Level Driver Library	"l_dsp.h/oba/obc", "l_edac.h/obc", "l_eeprom.obc", "l_errcod.h", "l_gendef.h", "l_hwmap.h", "l_itlmis.oba", "l_memory.h/oba/pbc", "l_pscgen.obc", "l_pscint.h/obc" "l_smcsco.h/obc", "l_smcsge.h/obc", "l_smcsin.h/obc", "l_smcsre.h/obc", "l_smcstr.h/obc"
Communication Interface	"datatx.c", "smcs_isr.s", "spu_io.c", "supervs.c", "watchpc.c"
Command Acknowledgment	"C2EEPROM.c", "check.c", "DMC_Con.c" "dump.c", "Load.c", "perform.c", "Rc_Sel.c", "Spu_tst.c", "Str_Stp.c", "w_reset.c", "write.c"
Compression Software	"average.c", "bitmodel.c", "bol1_2.c", "bol3.c", "bol4.c", "bol_ex.c", "buf_bol.c", "buf_spec.c", "csw.c", "dmch_cp.c", "dxs.c", "fill_in.c", "fill_out.c", "integ.c", "llc.c", "pacs_cod.c", "pacs_srt.c", "p_proc.c", "qsmodel.c", "ramp_ft.c", "rangecod.c", "spec1_2.c" , "spec3.c" , "spec4.c", "T_S_fm.c", "T_S_Red.c"
Housekeeping	"hk.c"
Miscellaneous	"phot_gn.c", "spec_gn.c", "Spvs_Tst.c", "asm_lib.s"
Additional	"Node1.c", "Node1.h", "allnodes.h"

Table 4. Software Modules of the SPU HLSW

4.2. Input Specifications

In this section the input specifications for the SPU HLSW are described. There exist two different input types:

- Commands from DPU are described in section 4.2.1.
- Science data with header from DEC/MEC. The header is described in section 4.2.2.

4.2.1. DPU Commands

In Table 5 you can find a description of all supported DPU commands addressed to the SPU HLSW. The packet structure of the 20 different DPU commands is given in AD002.

Command		Comments
LOAD		write data on the SPU memory
DUMP		read data from the SPU memory
CHECK		check specified SPU memory locations.
Perform Activity	CP_DATA_RAM_EEPROM	It burns the upgraded HLSW into EEPROM.
	RESET	performs SPU sub-unit warm reset
	RAW_CHAN_TRAN_MODE	Selects the number and index of raw channel
	STOP_REDUCT_COMPR	interrupts the application SW
	START_REDUCT_COMPR	runs the application SW
	ACT_TEST_PHOT	generate synthetic data in photometry, compress it and send it to DPU
	ACT_TEST_SPEC	generate synthetic data in spectroscopy, compress it and send it to DPU
	DMC_CONNECT	start the SPU link to DEC/MEC as slave or master
Write	WRT_DXSx	write the tables of the selected detectors (x = 1 .. 7)
	WRT_DET_CST_SPEC	write the table of the photoconductors constants
	WRT_DET_CST_PHOT	write the table of the bolometer constants
	WRT_SIM_DATA	write the table of parameters

Table 5. Description of the SPU HLSW Commands

4.2.2. DEC/MEC Science Data and Header

The size of the DEC/MEC header field is 64 Bytes for spectroscopy and for photometry. The DEC/MEC science data are attached to this header. Their associated fields are described in AD001.

4.3. Output Specifications

In this section the output specifications for the SPU HLSW are described. There exist three different output types:

- Command Acknowledgement, described in section 4.3.1.
- Housekeeping, described in section 4.3.2.
- Compressed Science Data, described in section 4.3.3.

4.3.1. Command Acknowledgement

Two kinds of DPU command acknowledgement are proposed: positive and negative. Their structures are described in AD001.

The positive acknowledgment informs that the SPU HLSW has received and understood the command (it is not sent after the execution of the command).

The SPU HLSW acknowledges the command reception and/or execution by sending the command response within a maximum timeout of 200 milliseconds.

4.3.2. Housekeeping

The size of SPU HK data field is 72 Bytes per board. In 5.4.1, Table 17 shows a detailed description of the individual fields.

When no raw data from DEC/MEC are received, only the CI and the PIX fields are filled. The other fields are not representative.

4.3.3. Compressed Data

The compressed data is sent from SPU to DPU within data packets described in AD001. Table 6 shows a short description of the fields of the compressed entity, which includes the compressed science data.

Description	Size [Bytes]	Compressed Science Data Field
Compressed Entity Header	28	CEH
Compressed DEC/MEC Header	<CDHS> 4	CDH
Compressed Science Data	<SCIS> 4	CSD

Table 6. Short Description of the Compressed Entity

1.) Compressed Entity Header is described in Section 5.5.1.

2.) Compressed DEC/MEC Header part contains the lossless compressed DEC/MEC Header.

3.) Compressed Science Data part contains:

a. Compressed Science data from the selected detectors. In default compressed mode, the obtained data after decompression, represent slopes of the sub-ramps in spectroscopy and 4-sample averages in photometry. The sub-ramp length in spectroscopy is set via the Write detector constants in spectroscopy command (see Section 5.1.1)

b. Lossless compressed Raw Channel data (if selected)

4.4. SMCS Chip Handling by SPU HLSW

4.4.1. SPU Handover to HLSW

The SPU HLSW is configured to start the DPU Link as Master (see link Start-up Protocol in RD005). When the SPU is handed over the HLSW, the following steps is performed (see more details in AD001) :

1. Reset SMCS chip
2. Set the DEC/MEC Link Connection Status in the HK to OFF
3. Set Nominal Configuration of the SMCS
4. Set DSP Interrupt at IRQ2 to Signal SMCS Events
5. Configure SMCS sub-interrupt for the two links (DPU and DMC). Unmask the following sub-interrupt (CH1_PAR_DIS_ERR, CH1_DATA_TXED, CH1_EOP_RXED, CH2_PAR_DIS_ERR, CH2_DATA_TXED and CH2_EOP_RXED)
6. Start the link to DPU as Master
7. Wait 9 seconds

8. If Connection to DPU is not established, then go again to Step 1. Otherwise
9. Start the Application SW tasks
    a. Communication Tasks and HK
    b. Compression Tasks
    c. Tasks for SPU Test Mode

The link start-up to DMC is performed under request using the command CONNECT_DMC. The start-up protocol (Master/Slave) could be chosen by the user. (see description in Section 5.1.1)

The DEC/MEC Link Connection Status in the HK is set to ON, whenever the connection between SPU and DMC is established

4.4.2. Connection Loss between SPU and Other Subunits

4.4.2.1. DPU-SPU Disconnect Error

When the SPU HLSW detects a disconnect error on the DPU link, the following steps are performed:

1. Stop the Application SW tasks
    a. Communication Tasks and HK
    b. Compression Tasks
    c. Tasks for SPU Test Mode
2. Perform the Steps 1-9 in Section 4.4.1

4.4.2.2. DMC-SPU Disconnect Error

When the SPU HLSW detects a disconnect error on the DMC link, the DMC Link Status in the HK is set to OFF (0x00).

5. Reference

5.1. DPU commands

5.1.1. Parameters and Examples

In this section you can find a detailed description and declarations for the HLSW command usage. In the following table, a summary of all the HLSW supported DPU commands are listed including the command structure and parameters.

☞ Note that the SAU is different among DRAM/PRAM/DPRAM. The SAU's are defined in RD011.

Command	Command Structure and Parameters						
	Field 1 CommandID 2 Bytes	**Field 2** 2 Bytes	**Field 3** 2 Bytes	**Field 4** 2 Bytes	**Field 5**	**Field 6** 2 Bytes	**Field 7** 2 Bytes
LOAD	(0x01)	Adress1	Adress2	Length	Data <Length>SAUs	Checksum	
DUMP	(0x02)	Adress1	Adress2	Length			
CHECK	(0x03)	Adress1	Adress2	Length			
CP_DATA_RAM_EEPROM	(0x04)	Spare	Activity ID (0x04)	Structure ID (0x05)	Parameter 1-5 5 x 4 Bytes		
RESET	(0x04)	Spare	Activity ID (0x05)	Structure ID (0x00)			
RAW_CHAN_TRAN_MODE	(0x04)	Spare	Activity ID (0x06)	Structure ID (0x05)	Parameter 1-5 5 x 4 Bytes		
STOP_REDUCT_COMPR	(0x04)	Spare	Activity ID (0x07)	Structure ID (0x00)			
START_REDUCT_COMPR	(0x04)	Spare	Activity ID (0x08)	Structure ID (0x00)			
ACT_TEST_PHOT	(0x04)	Spare	Activity ID (0x0A)	Structure ID (0x00)			
ACT_TEST_SPEC	(0x04)	Spare	Activity ID (0x0B)	Structure ID (0x00)			
CONNECT_DMC	(0x04)	Spare	Activity ID (0x10)	Structure ID (0x01)	Parameter 1 4 Bytes		
WRT_DXS1	(0x06)	Spare	Parameter ID (0x81)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS2	(0x06)	Spare	Parameter ID (0x82)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS3	(0x06)	Spare	Parameter ID (0x83)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS4	(0x06)	Spare	Parameter ID (0x84)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS5	(0x06)	Spare	Parameter ID (0x85)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS6	(0x06)	Spare	Parameter ID (0x86)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DXS7	(0x06)	Spare	Parameter ID (0x87)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DET_CST_SPEC	(0x06)	Spare	Parameter ID (0x42)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_DET_CST_PHOT	(0x06)	Spare	Parameter ID (0x24)	Length (0x18)	Data 96 Bytes	Checksum	Spare
WRT_SIM_DATA	(0x06)	Spare	Parameter ID (0x18)	Length (0x18)	Data 96 Bytes	Checksum	Spare

Table 7 . Supported DPU commands

LOAD

Summary load data to the SPU memory

Return value positive acknowledgement

Structure

Command ID	Address1	Address2	Length	Data	Checksum
2 Bytes	2 Bytes	2 Bytes	2 Bytes	<Lenght words>	2 Bytes

Description

This command is used to write data on the SPU DRAM and DPRAM. See document AD002 for further details.

☞ See AD002 for the detailed definitions of the Address fields to address the memory blocks in a correct way:

☞ The Checksum field contains the checksum of the Data field.

☞ Negative acknowledgment is generated for these errors “invalid Memory ID”, “invalid Address”, “invalid Checksum” or “invalid Checksum after re-reading” (see also section 5.3.1).

The acknowledgement of this command is sent after the execution of the command.

Pay attention to not load data into the memory segments seg_dmda and seg_stak listed in the Software Architecture description Appendix A.

This command will not check the length of the Data field with the value declared in the Length field.

The HLSW does not load to the PRAM and to EEPROM.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the LOAD command parameters
void Load(u4 buf[])
```

DUMP

Summary read data from the SPU memory

Return values positive acknowledgement (data, checksum)

Structure

Command ID	Address1	Address2	Length
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

This command is used to read data from the SPU DRAM, PRAM and DPRAM. See document AD002 for further details.

☞ See AD002 for the detailed definitions of the Address fields to address the memory blocks in a correct way:

☞ Negative acknowledgment is generated for these errors “invalid Memory ID” or “invalid Address” (see also section 5.3.1).

The acknowledgement of this command is sent after the execution of the command.

This command will not check the Length field.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the DUMP command parameters
void Dump(u4 buf[])
```

CHECK

Summary checks specified SPU memory locations

Return value positive acknowledgement (checksum)

Structure

Command ID 2 Bytes	**Address1** 2 Bytes	**Address2** 2 Bytes	**Length** 2 Bytes

Description

This command is used to check specified SPU memory locations. See document AD002 for further details.

☞ See AD002 for the detailed definitions of the Address fields to address the memory blocks in a correct way:

☞ Negative acknowledgment is generated for these errors “invalid Memory ID” or “invalid Address” (see also section 5.3.1).

The acknowledgement of this command is not sent after the execution of the command.

This command will not check the Length field.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the CHECK command parameters
void Check(u4 buf[])
```

CP_DATA_RAM_EEPROM

Summary Copies Data from RAM to EEPROM

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID	Parameter1..5
2 Bytes	2 Bytes	2 Bytes	2 Bytes	5x4 Bytes

Description

After the reception of this command the SPU HLSW will burn the RAM content into the EEPROM. See document AD001 for further details.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID, invalid Structure ID or invalid Parameter) a negative acknowledgment is generated (see also section 5.3.1).

☞ The time needed for the execution of the command is about 4 seconds (according to RD010).

✋ The HK transmission to DPU is interrupted while executing the command.

✋ The acknowledgement of this command is sent before the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the CP_DATA_RAM_EEPROM command parameters
void C2EEPROM(u4 buf[])
```

RESET

Summary performs a warm reset of the SPU sub-unit

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

The SPU sub-unit will be reset whenever this command is received. It performs a warm start of the SPU preserving the content of the RAM. See document AD002 for further details.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

The acknowledgement of this command is not sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the RESET command parameters
void W_Reset(u4 buf[])
```

RAW_CHAN_TRAN_MODE

Summary: Selects the number and index of raw channel in photometry and in spectroscopy

Return value: positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID	Parameter1..5
2 Bytes	2 Bytes	2 Bytes	2 Bytes	5x4 Bytes

with the following parameters:

Spare	Observing Mode	Number of RC	RC Index	Spare
4 Bytes	4 Bytes	4 Bytes	4 Bytes	4 Bytes

Description

After the reception of this command the SPU HLSW will set the new Raw Channel Selection parameters (number of selected raw channel and initial index). The raw channel information is used while the HLSW is started for data compression. The ASW transmits data from selected channels, compressed lossless, within the compressed science data. See document AD001 for further details.

The two spares in the parameter section are ignored, observing mode is 1 (spectroscopy) or 2 (photometry), the number of additional raw channels is one of 0..31 and the RC index specifies the start index (1..array size). The indices take detector selection into account.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID, invalid Structure ID or invalid Parameter) a negative acknowledgment is generated (see also section 5.3.1).

☞ In spectroscopy it is possible to inhibit the rotating of the raw channels. In this case it is also advisable to choose a meaningful initial index, but this can be tricky: these indices are in the chaotic order in which the SPU receives the data!

✋ The acknowledgement of this command is sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the RAW_CHAN_TRAN_MODE command parameters
void Rc_Sel(u4 buf[])
```

STOP_REDUCT_COMPR

Summary stops the Application Software

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

This command is transmitted for the interruption of the Application Software.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the STOP_REDUCT_COMPR command parameters
void Stop(u4 buf[])
```

START_REDUCT_COMPR

Summary starts the Application Software

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

The SPU HLSW will run the Application Software after the reception of the start command. See document AD002 for further details.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the START_REDUCT_COMPR command parameters
void Start(u4 buf[])
```

ACT_TEST_PHOT

Summary generates and compresses simulated data in photometry

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

After the reception of this command the SPU HLSW will generate synthetic data in photometry, compress it and send it to DPU. See document AD002 for further details.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the ACT_TEST_PHOT command parameters
void phot_tst(u4 buf[])
```

ACT_TEST_SPEC

Summary generates and compresses simulated data in spectroscopy

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID
2 Bytes	2 Bytes	2 Bytes	2 Bytes

Description

After the reception of this command the SPU HLSW will generate synthetic data in spectroscopy, compress it and send it to DPU. See document AD002 for further details.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the ACT_TEST_SPEC command parameters
void spec_tst(u4 buf[])
```

CONNECT_DMC

Summary start the SPU link to DEC/MEC as slave or master

Return value positive acknowledgement

Structure

Command ID	Spare	Activity ID	Structure ID	Parameter1
2 Bytes	2 Bytes	2 Bytes	2 Bytes	4 Bytes

Description

This command is used to start the SPU link to DEC/MEC as slave or master. See document AD002 for further details.

Before to send this command, the items listed below should be taken into account:

1. HLSW should be in stopped mode (reduction/compression is not started)
2. DMC should be switched on with HLSW running
3. DMC-SPU HLSW connection should not be already established.
4. If DMC is Master of the DMC-SPU communication, the Parameter1 for the command should be 0x22
5. If DMC is Slave of the DMC-SPU communication, the Parameter1 for the command should be 0x11
6. If DMC is Slave of the DMC-SPU communication, DMC SW should have finished its reset cycle and already starting-up its link to SPU.

☞ If there is a Perform Activity command that cannot be understood (due to invalid Activity ID and invalid Structure ID) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent before the execution of the command.

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the CONNECT_DMC command parameters
void DMCCon(u4 buf[])
```

WRT_DXS1-7

Summary write tables of the selected detectors

Return value positive acknowledgement

Structure

Command ID	Spare	Parameter ID	Length	Data	Checksum	Spare
2 Bytes	2 Bytes	2 Bytes	2 Bytes	<0x18 words>	2 Bytes	2 Bytes

Description

This command is used to write the tables of selected detectors (from which data are requested) on the SPU memory. WRT_DXS1-5 are reserved for the bolometers while WRT_DXS6-7 are reserved for the photoconductors. In the following list the Parameter ID of the individual WRT_DXS commands are given:

Command	Parameter ID	Comments
WRT_DXS1	0x81	corresponds to the Array ID 1 of the blue detectors of Bolometer
WRT_DXS2	0x82	corresponds to the Array ID 2 of the blue detectors of Bolometer
WRT_DXS3	0x83	corresponds to the Array ID 3 of the blue detectors of Bolometer
WRT_DXS4	0x84	corresponds to the Array ID 4 of the blue detectors of Bolometer
WRT_DXS5	0x85	corresponds to the Array ID 5 of the red detectors of Bolometer
WRT_DXS6	0x86	corresponds in the blue/red SPU to the blue/red detectors of Photoconductors
WRT_DXS7	0x87	corresponds in the blue/red SPU to the degraded table (not used)

Table 8. List of Parameter IDs for the WRT_DXS command

The content of the Data field is the SW table to be written in the memory. The structure of this SW table is described in Table 9 (see also AD001).

☞ The Checksum field contains the checksum of the Data field.

☞ If there is a WRT_DXS command, which cannot be understood (due to incompatible Length with Parameter ID, invalid Checksum, invalid Parameter ID and invalid Checksum after re-reading) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

💣 This command will not check the length of the Data field with the value declared in the Length field.

Field	Description	Length (Bytes)	Comments
DXID	Detector Selection Table Identification	4	The DXID is a 32 bit integer. It is used to identify the table used for compression in order to decompress in a proper way. This information is also downlinked in the DECID field as part of the compressed entity header. The DXID is most certainly used like a counter. Default is 0x01 for all detectors selected
PAR ID	Parameter Identification	4	The PAR ID is a 32 bit integer that is not used by the HLSW. It is reserved for the on-ground decompression software.
DXSNB	Number of Selected Detectors	4	The DXSNB is an integer, which contains the number of selected detectors of the table mask.
DXMASK	Mask of Selected Detectors.	64 (Phot.) / 58.5 (Spec.)	The DXMASK is a bit mask setting detectors active with 1 and deactivating pixels with 0. For instance, setting the first 3 pixels active and the rest of the array blind is done with a mask containing the bits: 11100000… The order used is the general one described in the figures above. Note that there are 44 bits padded for alignment in spectroscopy.
DXSPARE	Spare Field	20 (Phot.) / 25.5 (Spec.)	The DXSPARE ends the detector selection table with zeros to maintain 32-bit granularity.

Table 9. Description of the Detector Selection Table Fields

The same HLSW is running in both SPUs Red and Blue. The SW accepts all seven write commands WRT_DXS1-7. The user has to choose the command to send depending on the Detector type (photometer WRT_DXS1-5 or spectrometer WRT_DXS6-7).

Both SPUs have the same set of commands, the SW accepts all commands but the usage of the tables depends on the observing mode (photometry or spectroscopy) and the wavelength (SWL and LWL).

- WRT_DXS1-5 commands are required for photometry data processing for both SPUs
- WRT_DXS6 command is actually required for spectroscopy for both SPUs
- WRT_DXS1-4 and WRT_DXS6 commands are required for SWL data processing in any SPU
- WRT_DXS5 and WRT_DXS6 commands are required for LWL data processing in any SPU
- WRT_DXS7 command may be used in degraded mode whenever one SPU sub-unit has to process data from both detectors (SWL and LWL). In the current SW version WRT_DXS7 is not exploited as the degraded mode is not implemented .

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the Write command parameters
void Write(u4 buf[])
```

Make sure to also provide valid parameters for the additional raw channels after having uploaded the detector selection tables.

WRT_DET_CST_SPEC

Summary write the table of the photoconductor constants

Return value positive acknowledgement

Structure

Command ID	Spare	Parameter ID	Length	Data	Checksum	Spare
2 Bytes	2 Bytes	2 Bytes	2 Bytes	<0x18 words>	2 Bytes	2 Bytes

Description

This command is used to write the table of the photoconductors constants, which are relevant for the pre-processing step.. The content of the Data field is the SW table to be written in the memory. The structure of this SW table should be as described in AD001. See also document AD002 for further details

- Whenever one detector constant is changed, the whole table has to be updated.
- The Checksum field contains the checksum of the Data field.
- If there is a WRT_DET_CST_SPEC command that cannot be understood (due to incompatible Length with Parameter ID, invalid Checksum, invalid Parameter ID and invalid Checksum after re-reading) a negative acknowledgment is generated (see also section 5.3.1).
- The acknowledgement of this command is not sent after the execution of the command.
- This command will not check the length of the Data field with the value declared in the Length field.

Field	Description	Length (Bytes)	Comments
PPF	Pre-Processing Flag	4	0…no PP, 1…addition, 2…subtraction, 3…multiplication, 4…division. **Default value is 0**
PPP	Pre-Processing Parameter	4	a 32-bit value used according to the PPF. **Default value is 0**
TFS	Threshold For Spectroscopy	4	This parameter is used for the glitch detection in a ramp. **Its value is 1000 by default.**
TCS	Threshold for Chopper position deviation in Spectroscopy	4	This parameter is used to distinguish between two chopper positions. **Its value is 100 by default.**
TGD	Threshold for Grating position deviation in Spectroscopy	4	This parameter is used to distinguish between two grating positions. **Its value is 100 by default.**

RSRR	Readout Per Sub_Ramp Readback	4	This parameter is used to set the number of samples to be fit per segmented ramp. **Its value is 32 by default.** Use powers of 2 here for ramp fitting: 2,4,…,512 Take care that this number should be equal or smaller than the number of samples per ramp (Readouts per Ramp Readback) Choose this number such the CPU workload is less than 100%. See Section5.7 for a typical CPU workload of this version of the SW
GDFS	Glitch Detection For Spectroscopy	4	This parameter defines the use or not of the glitch detection algorithm before ramp fitting. 0: Use glitch detection Otherwise: Do not use glitch detection (**default = 1**)
RFAL	Ramp Fitting ALgorithm	4	This parameter is used to choose the fitting algorithm. 0: Least squares algorithm (**default**) 1: Mean algorithm (**recommended**)
SOAL	Sorting Algorithm	4	The allowed values are: 0,1,2,3 and 4. The default value is 4. **Recommended value is 5.** 0: Spatial redundancy reduction 1: Differential signal + Spatial redundancy reduction 2: Reordering to ramp indexes + Spat. Red. reduction 3: Differential signal + Reordering to ramp indexes + Spatial redundancy reduction 4: Resorting the data MSBs-LSBs + Spatial redundancy reduction 5: rampDiff
LCAL	Lossless Compression ALgorithm	4	The allowed values are: 0..4. Initial default is 0. 0: pacs_codec 1: RZIP + pacs_codec 2: RZIP 3: Arithmetic coding 4: FM Ari – **recommended value**
PCOD	Pacs Codec OrDer	4	Allowed values are: 3 and 4. **Recommended is 3.** Take care that this number should not be lower than 3 (Invalid number) and not greater than 4 (High CPU workload)
RNDB	Number of Rounding Bits	4	Allowed values: 0, 1, 2, 3. **The default value is 0.**
RNDS	Rounding Strategy / Averaging Algorithm	4	Choose algorithm: 0: Roland, 1: Koryo, 2: v13.8. **The default value is 0.**
FIXRC	Fix Raw Channels	4	Set to 1 to inhibit the rotation of additional raw channels. **The default value is 0.**
DPRE	Discard PRE	4	**The default value is 0.** Sets how many samples at the beginning of a ramp are to be discarded. The values 0-7 can be chosen. They are effective in default mode when submeans are to be made with rounding strategy 0 or 1.
DPOST	Discard POST	4	**Default: 0.** Same as DPRE, but for the end of a ramp.
SPARE		32	32 Bytes left

Table 10. Description of the Detectors Constants for Spectroscopy Table Fields

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the Write command parameters
void Write(u4 buf[])
```

WRT_DET_CST_PHOT

Summary write the table of the bolometer constants

Return value positive acknowledgement

Structure

Command ID 2 Bytes	Spare 2 Bytes	Parameter ID 2 Bytes	Length 2 Bytes	Data <0x18 words>	Checksum 2 Bytes	Spare 2 Bytes

Description

This command is used to write the table of the bolometer constants, which are relevant for the pre-processing step. The content of the Data field is the SW table to be written in the memory. See also document AD002 for further details.

☞ Whenever one detector constant is changed, the whole table has to be updated.

☞ The Checksum field contains the checksum of the Data field.

☞ If there is a WRT_DET_CST_PHOT command that cannot be understood (due to incompatible Length with Parameter ID, invalid Checksum, invalid Parameter ID and invalid Checksum after re-reading) a negative acknowledgment is generated (see also section 5.3.1).

✋ The acknowledgement of this command is not sent after the execution of the command.

💣 This command will not check the length of the Data field with the value declared in the Length field.

Field	Description	Length (Bytes)	Comments
PPF	Pre-Processing Flag	4	0…no PP, 1…addition, 2…subtraction, 3…multiplication, 4…division. **Default: 0**
PPP	Pre-Processing Parameter	4	a 32-bit value used according to the PPF, **Default: 0**
TFP	Threshold For Photometry	4	This parameter is used for the glitch detection in photometry. **Its Value is 1000 by default.**
TCP	Threshold for Chopper position deviation in Photometry	4	This parameter is used to distinguish between two chopper positions. **Its value is 100 by default.**

GDFP	Glitch Detection For Photometry	4	This parameter defines the use or not of the glitch detection algorithm while averaging. If glitch detection is chosen, HLSW will count the number of glitches in a compressed entity based on TFP parameter and send it in HK packet (SPUx_SAMP_CORR). Glitches are not removed on-board. 1: Do not use glitch detection **(default)** Otherwise: Use glitch detection
NAVG	Number of Samples to Average	4	This parameter decides whether averaging is performed with an integer algorithm for a fixed number of samples (4 in default mode, 8 in double mode) or an arbitrary number of samples is used in a floating-point algorithm. 0: use compression mode (default=4, double=8, half=2). **This is the default.** x: use x samples for averaging. Meaningful values are: 2..120 (max averaging length).
RND	Rounding parameter	4	Parameter to set optional rounding. **0 (default)** means no additional rounding. The values 1, 2 and 3 can be set here to allow 1..3 bit rounding of the averages.
RNDS	Rounding Strategy / Averaging Algorithm	4	Choose algorithm: 0: Roland, 1: Koryo, 2: v13.8. **The default value is 0.**
SCM	Lossless Compression Scheme	4	The allowed values are: 0, 1. **Recommended is 1.** 0: “velveteen”, pre-fm scheme used in 12.8 1: “velvet”, fm scheme with enhanced compression
LCAL	Lossless Compr. Algorithm	4	**The default value is 4.** This parameter is only relevant for lossless only mode. (0x04)
PCOD	Pacs Codec OrDer	4	**The default value is 3.** This parameter is only relevant for lossless only mode (0x04)
DPRE	Discard PRE	4	**The default value is 0.** Sets how many samples at the beginning of an averaging group are to be discarded. The values 0-7 can be chosen. They are only effective in default mode and also only if RNDS is 0 or 1.
DPOST	Discard POST	4	**The default value is 0.** Same as DPRE, but for the end of an averaging group.
SDEL	Sync Delay	4	**The default value is 0.** This delays the start of a label sequence by the number of frames set here. Allowed numbers are 0..3. Effective in all modes except BTM!
SPARE	Spare	40	40 Bytes left

Table 11. Description of the Detectors Constants for Photometry Table Fields

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the Write command parameters
void Write(u4 buf[])
```

WRT_SIM_DATA

Summary write the table of parameters for the SPU test mode

Return value positive acknowledgement

Structure

Command ID	Spare	Parameter ID	Length	Data	Checksum	Spare
2 Bytes	2 Bytes	2 Bytes	2 Bytes	<0x18 words>	2 Bytes	2 Bytes

Description

This command is used to write the table of parameters, which will be used, for the SPU test mode. The content of the Data field is the SW table to be written in the memory. See also document AD002 for further details.

☞ Whenever the to-be-simulated data should be changed, the whole table has to be updated.

☞ The Checksum field contains the checksum of the Data field.

☞ If there is a WRT_SIM_DATA command that cannot be understood (due to incompatible Length with Parameter ID, invalid Checksum, invalid Parameter ID and invalid Checksum after re-reading) a negative acknowledgment is generated (see also section 5.3.1).

The acknowledgement of this command is not sent after the execution of the command.

This command will not check the length of the Data field with the value declared in the Length field.

Field	Description	Length (Bytes)	Comments
SPUID	SPU Identifier	4	Identifies the channel (red or blue SPU). - 0x02: for LWL SPU (red) - 0x03: for SWL SPU (blue) (the default value on-board)
CMM	Compression Mode	4	Defines the compression mode of the HLSW. - 0x00-0x02, 0x04, 0x07, 0x09: for Compression Modes in Photometry (see Table 14) (0x00 is the default value on-board) - 0x10,0x11,0x14, 0x17–0x19: for Compression Modes in Spectroscopy (see Table 14) (0x10 is the default value on-board)
RRR	Readouts in Ramp Readback	4	Number of Samples per Ramp/Average. (32/4 is the default value for spectroscopy/photometry)
SPARE	Spare	15*4	60 Spare bytes.
HCMP	Header Compression Algorithm	4	Choose between new FM (0x0) and old (0x1) header compression. Default is 0.

CKSM	Checksum protection for compressed entity	4	0: do not use checksum protection 1: use checksum protection (default)
PAHK	Parameter Address for the HK	4	Physical address of the write parameter to be monitored. See the list of addresses in Table 25. Though the best source for addresses is the hlsw.map generated during compilation. The default value is 0xFFFFFFFF. Take care to choose a valid address value according to the given Table, otherwise the SW will not provide a value.
SPARE	Spare	12	12 Bytes left.

Table 12. Description of the Simulated Data Parameters Table Fields

Syntax

```
// #define u4 unsigned int
// u4 buf[1024] : contains the Write command parameters
void Write(u4 buf[])
```

5.2. DEC/MEC header and science data

5.2.1. Structure and Parameters

The structure of the DEC/MEC header and science data is described in Table 13.

word #	Photometry				Spectroscopy			
	Field description		Useful Bytes	Comments	Field description		Useful Bytes	Comments
1	SPUID (SPU Identification)		4	2 (red SPU) 3 (blue SPU)	SPUID (SPU Identification)		4	2 (red SPU) 3 (blue SPU)
2	Type		1	0x02 (Phot.)	Type		1	0x01 (Spec.)
3	OBSID (Observation ID)		4		OBSID (Observation ID)		4	
4	BBID (Building Block Identific.)		4		BBID (Building Block Identific.)		4	
5	LBL (Label)		1	see Note below	LBL (Label)		1	see Note below
6 7	TMP (Timing Parameters)		6		TMP (Timing Parameters)		6	
8	ROSP (ReadOut Specifications in Photometry)	VLD (Validity)	1	0xff (valid) 0x00 (invalid) see Note below	ROSS (ReadOut Specifications in Spectroscopy)	VLD (Validity)	1	0xff (valid) 0x00 (invalid) see Note below
9		CPR (Chopper Position Readback)	3			CPR (Chopper Position Readback)	3	
10		WPR (Wheel Position Readback)	1			WPR (Wheel Position Readback)	1	
11		BOLST (BOLC Status)	2			GPR (Grating Position Readback)	3	
12		CRDC (5 times OBT clock ticks since the last SET_TIME)	3			CRCRMP (Current Readout Count in a RaMP)	2	RRR-1 to 0
13		CRDCCP (Current ReaDout Count in Chopper Position)	1			RRR (Readouts in Ramp Readback)	2	2^n, n = 3 to 10
14		DBID (Data Block ID)	1	see AD001 for details		CRDC (Nb of Readouts since the last SET_TIME)	3	
15	BSID (Bolometer Setup Identification)		1	Information for the Bolometers		CRECR (CRE Control Readback)	2	
16	Compression Parameters		2	see Note below and Table 14	Compression Parameters		2	see Note below and Table 14
17+	Science data (256 words)				Science data (234 words)			

Table 13. Structure and Parameters of the DEC/MEC header and science data

Label (LBL)

The Label statement is meant to indicate to the SPU-HLSW across which "quadruples" it has to average in photometry. It is further used also to indicate the number of discrete chopper steps within a sequence. In spectroscopy the Label statement is not used. Hereafter, the following conventions for the Label in photometry:

If Label is set to 0, then no sequence is active otherwise

Bit 1 … is set to 1, to indicate that DMC sequence is active.

Bit 2-6 … counts chopper positions within sequence
Bit 7 … signal from BB1
Bit 8 … signal from BB2

LABEL = 63 has a special meaning and indicates "freeze frame mode"

Validity (VLD)

The "Validity" statement defines if the data is valid or not. This field is not checked by HLSW for photometry' data.

If spectroscopy data is declared invalid, the data will be ignored and the buffer is cleaned until valid data is arriving.

Compression Parameters

The compression and transmission mode which corresponds to the Compression Parameters field (CMM) in Table 13 are listed in Table 14.

CMM	Photometry	Spectroscopy
0x00	BOL0: Default Mode	None
0x01	BOL1: Double Compression Mode	None
0x02	BOL2: Half Compression Mode	None
0x04	BOL4: Lossless Compression Mode	None
0x07	BOL7: Transparent Mode	None
0x09	BOL9: Buffer Transmission Mode	None
0x0c	CS: Compressed Sensing Mode	None
0x10	None	SPEC0: Default Mode
0x11	None	SPEC1: Default Mode (same as SPEC0)
0x14	None	SPEC4: Lossless Compression Mode
0x17	None	SPEC7: Transparent Mode
0x18	None	SPEC8: Compression Mode for 4 second reset
0x19	None	SPEC9: Buffer Transmission Mode

Table 14. List of Compression Parameters for Photometry and Spectroscopy

It is recommended to start the Buffer transmission mode from a stopped compression state. Transition from a running compression mode to the Buffer transmission mode without stopping the compression software in between generally works, but may have unpredictable side-effects.

It is mandatory to stop the compression software when running the Buffer transmission mode before to switch to another compression mode. Transition from the Buffer transmission mode to another compression mode without stopping the compression software in between does not switch at all.

The maximum number of detectors to select in lossless compression mode are 60 in spectroscopy, 256 in SWL photometry and 64 in LWL photometry, for complexity reason. If more than those numbers are commanded the SPU HLSW will transmit only data from the maximum allowed number of detectors.

The maximum number of samples per subramp to choose in the compression mode for 4 second reset is 512 samples.

Compressed Sensing mode is a prototype so far. Its datarate is by far too high and there is no Decompression available yet.

In this SW version, the spectroscopy' double compression mode is equivalent to the spectroscopy' default mode.

5.3. Command Acknowledgement

5.3.1. Structure and Error Codes

Two kinds of acknowledgement are proposed: positive and negative. Their packet structures are represented respectively in Table 15. The error code and information of the attached parameters of the negative acknowledgement can be found in Table 16.

Command	2 Bytes	2 Bytes	2 Bytes	2 Bytes	N x 2 Bytes	2 Bytes
Load	PACK ID 0x81					
	NACK ID 0xF1	Error Code See Table 16	Parameter See Table 16			
Dump	PACK ID 0x182* 0x082**	Address1	Address2	Length	Data	Checksum
	NACK ID 0xF2	Error Code see Table 16	Parameter See Table 16			
Check	PACK ID 0x83	Address1	Address2	Length	Checksum (2 Bytes)	
	NACK ID 0xF3	Error Code see Table 16	Parameter See Table 16			
Perform Activity	PACK ID 0x84					
	NACK ID 0xF4	Error Code see Table 16	Parameter See Table 16			
Write command	PACK ID 0x86					
	NACK ID 0xF6	Error Code see Table 16	Parameter See Table 16			
Invalid command	NACK ID 0x1FF	Error Code see Table 16	Parameter See Table 16			

* ... Intermediate packets
** ... Last packet

Table 15. Packet Structure of SPU HLSW Command Acknowledgements

Error Code	Error Definition	Parameter attached …	Load	Dump	Check	Perform	Write	invalid
0x71	invalid command ID	Command ID (4 Bytes)						•
0x72	invalid Memory ID	Spare (3 Bytes) and Memory ID (1 Byte)	•	•	•			
0x73	invalid Address	Spare (1Byte) and Address (3 Bytes)	•	•	•			
0x74	Length not compatible to Parameter ID	Spare (2 Bytes) and Length (2 Bytes)					•	
0x75	invalid Checksum	Spare (2 Bytes) and Calculated Checksum (2 Bytes)	•				•	
0x76	invalid Parameter ID	Spare (1Byte) and Address (3 Bytes)					•	
0x77	invalid Activity ID	Spare (2 Bytes) and Activity ID (2 Bytes)				•		
0x78	invalid Structure ID	Spare (2 Bytes) and Structure ID (2 Bytes)				•		
0x79	invalid SID Parameter[1]	Parameter (4 Bytes)				•	•	
0x7B	invalid Checksum after re-reading	Spare (2 Bytes) and Checksum after re-reading (2 Bytes)	•				•	

Table 16. Error Codes and Attached Parameters

[1] Invalid SID parameter means that the parameter is not in a valid range according to command description.

5.4. Housekeeping

5.4.1. Structure and Parameters

In Table 17 you can find a detailed list of HK parameters with the nominal values, the limits and description.

Parameter	Size (B)	Variable	Default Value	Description
SPUS_HK_HEADER SPUL_HK_HEADER	4	HK Header	0x00870000	This field is added to the HK packet sent from SPU to DPU. It distinguishes the HK from the science packet and the response packet
SPUS_OBSID SPUL_OBSID	4	Observation Identification	0	This field is used to identify the observation the HK belongs to. It should be the same OBSID for the science data.
SPUS_PIX SPUL_PIX	4	Packet Index	0xFFFFXXXX	- The two MSB of the PIX are set to 0, when the HK packet contains the compression results (compression started). In this case, the two LSB are counting the number of downlinked compressed entities since the HLSW handover of the SPU or the last warm reset command. Moreover, the 2 LSB provide the relationship between the science packet and the related HK. The concordance is determined on ground. - The two MSB of the PIX are set to: 1.) 0xFFFF for stopped compression, 2.) 0xDDDD for started compression or 3.) 0xAAAA for "compressed entity incomplete", signalling a transition or loss of data due to high cpu load. - In all three cases, the two LSB are stalled to the last updated value It is very important to start compression from 0xFFFFXXXX state.

SPUS_CI SPUL_CI	2	Counter Increments	0xXXXX	This field is used to attest that HLSW is not hanging. It is periodically incremented from 0 to 65535.
SPUS_REAL SPUL_REAL	2	Reduction Algorithm Used	0xFF	Information about the used reduction algorithm, including compression scheme and additional rounding. It is set to 0xFF if on-board data reduction is not used.
SPUS_SATUR_FLAG SPUL_SATUR_FLAG	1	Saturation Flag	0	0x11: Saturated, 0: Non-Saturated
SPUS_SAMP_CORR SPUL_SAMP_CORR	3	Glitch Counter Information	0	Number of samples affected by glitches for the blue SPU.
SPUS_MAINT_RAMPS SPUL_MAINT_RAMPS	2	Number of Maintained Ramps/ Averages	0	Number of maintained slopes or sub-slopes (Spectroscopy) or averages (Photometry).
SPUS_CPU_WORKLOAD SPUL_CPU_WORKLOAD	2	CPU Workload	0xXXXX	CPU workload for the past second. It is represented in units of one-tenth of a percent as integer.
SPUS_DMC_LINK_STATUS SPUL_DMC_LINK_STATUS	2	DEC/MEC Link Status	0xFF	DEC/MEC Link Status: 0x00: Link is OFF, 0xFF: Link is ON and 0xAA: Link connection is in progress.
SPUS_INTEG_RAMPS SPUL_INTEG_RAMPS	1	Number of Integrated Ramps/ Averages	0	Number of integrated ramps (Spectroscopy) or averages (Photometry).
SPUS_VID SPUL_VID	1	Version ID	0xXX	This field is used to represent the software version
SPUS_RCX SPUL_RCX	2	Raw Channel Index	0	The most significant 6 bits contain the Nb. of selected raw channels (RCNB) The least significant 10 bits contains the Index of Raw Channels (RCX) until which RCNB pixels data are transmitted with the compressed science data.
SPUS_DMC_ERROR SPUL_DMC_ERROR	1	DEC/MEC Header Error	0	This field is set to a message code if an error is detected in the DEC/MEC header. See the list of errors in Table 19
SPUS_MEM_STATUS_CNTS SPUL_MEM_STATUS_CNTS	2	Memory Status Counters	0xXXXX	LSB: EDAC memory check counter for Single Error Failure MSB: EDAC memory check counter for Double Error Failure

SPUS_SUBVERSION SPUL_SUBVERSION	2	Sub-version Number	8	This number indicates the Sub-version. In release 13.5 this is 5.
SPUS_LLC_ERROR SPUL_LLC_ERROR	2	Invalid LLC Parameter	0	This HK parameter reports an invalid write parameter for LLC. Default = 0. See the actual HLSW user manual for more details
SPUS_PAR_MONITOR SPUL_PAR_MONITOR	2	Monitored Write Parameter	0	This HK parameter can be exploited by the user in monitoring a dedicated parameter in the SPU memory. The address of this parameter can be set via a write command (write simulated data parameter). The default value is a variable monitoring the buffer usage. See Table 25 for the physical address ranges.

Table 17. SPU HLSW HK List

5.4.2. DEC/MEC Header Errors List

The table below presents the codes and descriptions for the DMC header error that is flagged in SPU HK.

Error Code	Description
0x02	**Case Photometry or Spectroscopy:** transition buffer due to a CMM or OBSID change is ignored because its size is smaller than the minimum required granularity (see AD001).
0x04	**Case Photometry:** wrong SPUID (expected value 2 or 3).
0x05	**Case Photometry:** DBID is out of the range [0,5].
0x06	**Case Red Photometry:** SPUID (expected value 3) is not compatible with DBID (expected value 5).
0xAA	**Case Photometry or Spectroscopy:** wrong compression mode.
0xDD	Unable to process data fast enough due to high CPU load. The decompressed data stream will have gaps.
0xFF	Wrong observing mode (expected type values are 1 or 2)

Table 18. DMC Header Error Codes Descriptions

5.4.3. Monitoring the Memory Errors in the SPU HK

Four kinds of memory errors can be detected by the EDAC chip installed within the DSP board:

- Single Error Failure (SEF) in DM
- Single error failure in PM (PRAM and EEPROM)
- Double Error Failure (DEF) in DM
- Double error failure in PM (PRAM and EEPROM)

EDAC can only correct the single error failure.

HLSW uses the HK parameter SPUS_MEM_STATUS_CNTS to count the SEF (in LSB) and DEF (in MSB) occurred in SPU memory. This HK parameter is generated within every HK in SPU idle mode and after every 256 HK packet in SPU started mode (while compression started).

Each time ground people detect that one of the indices has incremented, it shall request a dump of the failing address array to know which cell has produced the error. If the same address is repeated many times in the array, it means that is has a permanent error.
The starting addresses for the arrays where the failing memory are stored are listed below.

Error Type	Memory ID	Starting Address	Length in DM words
DM SEF	Ext DRAM	0x4027FE00	128
DM DEF	Ext DRAM	0x4027FE80	128
PM SEF	Ext DRAM	0x4027FF00	128
PM DEF	Ext DRAM	0x4027FF80	128

5.5. Compressed Science data

5.5.1. Structure and Parameters

The Parameters which are used for the compressed entity are summarized in Table 19.

Word #	Field Description			Size (Bytes)	Comments
1	CEH (Compressed Entity Header)	Type		4	0x01 in spectroscopy 0x02 in photometry
2		PIX (Packet Index)		4	
3		DECID (DEcompression Code ID)	REAL (Reduction ALgorithm) . See below for the algorithms codifications	2	
			RCX (Raw Channel indeX). Same definition as RCX in HK (Table 17)	2	
4			Spare	2	
			Version ID	1	
			CMM (Compression Mode)	1	
5			DXS ID (Detector Selection Table ID)	4	
6		CRCS(Compressed Raw Channel data Size)		2	
		CDHS (Compressed DEC/MEC Header Size)		2	
7		SCIS (Science Data Size)		4	
8+	CDH (Compressed DEC/MEC Header)			<CDHS> 4	
8 + <CDHS> +	CSD (Compressed Science Data)			<SCIS> 4	

Table 19. Parameters of the Compressed Entity

The REAL field can contain the following codes:

Code	Description
0x04	Reduction in photometry: Mean algorithm for 4 samples (quadruple) is used
0x08	Reduction in photometry: Mean algorithm for 8 samples is used
0x82	Reduction in photometry: Mean algorithm for 2 samples is performed using the float algorithm.
0x8N	Reduction in photometry: Mean algorithm for N samples is performed using the float algorithm.
0xff	No reduction is performed. This code is used for modes with raw data only.
0x10	Reduction in spectroscopy: Least Square fit algorithm with glitch detection is used
0x11	Reduction in spectroscopy: Least Square fit algorithm without glitch detection is used
0x12	Reduction in spectroscopy: Mean algorithm with glitch detection is used
0x13	Reduction in spectroscopy: Mean algorithm without glitch detection is used
0x14	Reduction in spectroscopy: Two sample difference algorithm without glitch detection is used

Table 20. Codes of REAL Field

In addition to the values listed in the table, the software transmits information about the number of rounding bits in the high byte of the REAL field.

☞ The TM buffer capacity is 257 packets in each SPU board. Science packet might be lost between SPU and DPU if a compressed entity represents more than 257 packets.

☞ In case of LWL photometry, the SPU SW sends the TM packets in chunks every 4 seconds to avoid the buffer overflow in DPU for a huge number of science packets.

5.5.2. Data Rates of HLSW Compression Modes

The data rates of HLSW Compression Modes in Photometry are listed below. See AD001 for more details.

Compression Mode in Photometry	Description	Selected Detectors		Expected TM Rate [kbits/s]			Packets per second
		SPU LWL	SPU SWL	SPU LWL	SPU SWL	Total	
Mode 0	BOL0: Default Mode	512	2048	100	25	**125**	15.9
Mode 1	BOL1: Double Compression Mode	512	2048	60	15	**75**	9.6
Mode 2	BOL2: Half Compression Mode	512	2048	100	25	**125**	15.9
Mode 4	BOL4: Lossless Compression Mode	512 148	2048 592	323.39 94.53	81.89 24.67	**405.28** **119.20**	51.57 15.17
Mode 7	BOL7: Transparent Mode	512 37	2048 148	1289.11 94.48	323.28 24.63	**1612.39** **119.11**	205.15 15.16
Mode 9	BOL9, BUF_Bol: Buffer Transmission Mode	512	2048	94.50	118.60	**213.10**	27.11

Table 21. Expected Data Rates for the Compression Modes in Photometry

The data rates of HLSW Compression Modes in Spectroscopy are listed below. See AD001 for more details.

Compression Mode in Spectroscopy	Description	Selected Detectors		Expected TM Rate [kbits/s]			Packets per second
		SPU LWL	SPU SWL	SPU LWL	SPU SWL	Total	
Mode 0	SPEC0: Default Mode	450	450	58.76	58.76	**117.52**	14.95
Mode 1	SPEC1: Double Compression Mode	450	450	58.76	58.76	**117.52**	14.95
Mode 4	SPEC4: Lossless Compression Mode	450	450	454.88	454.88	**909.75**	115.75
		57	57	59.53	59.53	**119.06**	15.14
Mode 7	SPEC7: Transparent Mode	450	450	1812.94	1812.94	**3625.88**	461.34
		14	14	58.46	58.46	**116.92**	14.88
Mode 8	SPEC8: Compression Mode for 4s Reset	450	450	58.73	58.73	**117.47**	14.95
Mode 9	SPEC9, BUF_Spec: Buffer Transmission Mode	450	450	122.71	122.71	**245.42**	31.23

Table 22. Expected Data Rates for the Compression Modes in Spectroscopy

The data rate for one raw channel in spectroscopy is

256 Hz x 16 bit = **4 kbits/s**

and for one raw channel in photometry

40 Hz x 16 bit = **0,625 kbits/s**.

5.6. SPU Simulated Data

5.6.1. Spectroscopy

The data generator in spectroscopy simulates the detector readouts of one frame by the same value. The ramp has the following function:

1st sample =0x100+RRR
2nd sample = 0x100+RRR -1
3rd sample = 0x100+RRR – 2
…
last sample = 0x101

The result of the ramp-fitting module using the least squares algorithm is RRR x slope (in this case –RRR).

5.6.2. Photometry

The data generator in photometry simulates the detector readouts of one frame by the same value. The data has the following form:

0x0100
0x0100
0x0100
0x0100
0x0200
0x0200
0x0200
0x0200
0x0200
0x0200
0x0200
0x0200
0x0200
0x0100
0x0100
…

The expected average is a sequence of 0x0100 , 0x0200, 0x0200, 0x0100 ...

5.7. Typical CPU Workload and TM Rate

This Section describes the typical CPU workload and data rate for test data.

5.7.1. Spectroscopy

The compression software has been tested on the data derived from FM tests. In Spectroscopy, a datafile from FM-ILT has been assembled. It originates from data taken in Buffer Transmission Mode, *FILT_Spec_SPU_test_20061205_03.tm*.

The results for default modes from the test plan are shown in Table 23.

Nb. Of Samples per Ramp	Nb of Samples per Sub-mean	CPU Workload	Science Data TM Rate Kbits/s (red+blue)
64	32	55	70
64	16	80	130

Table 23. Typical CPU Workload and Data Rates for the current SW Version. Note that the CPU overall workload is given with no additional raw channels selected. Add 5% CPU load and 7 kbits/s for each additional raw channel.

5.7.2. Photometry

The compression software has been tested on the data derived from FM tests. Two test data files are used for photometry, phot1.dat.staring and phot1.dat.chopping. These files have unsigned data from the blue bolometer, originating from the files *TM2006_05_04__19_04_59brut.tm* and *TM2006_04_22__05_43_44brut.tm* taken by CEA. There is also another datafile with signed data taken on 2006-03-14, which is used for verification of the functionality and not used for performance measures.

Nb. Of Samples to average	Parameter for compression scheme	CPU Workload	Science Data TM Rate Kbits/s (red+blue)
4	0 (12.81 setting)	55	145
4	1	85	125

Table 24. Typical CPU Workload and Data Rates for the current SW Version. Two additional raw channels are included.

5.8. Important Physical Addresses in the Actual HLSW Version

This address list can be used by the user while choosing an address for watching a write parameter. The start and end addresses of the 3 write tables are listed in the Table below.

If the user needs to monitor the first parameter in e.g. write detector constants in spectroscopy table, then the address to choose if the start address + index –1 (index =1 in this case).

If the user needs to monitor the 10th parameter in e.g. write detector constants in spectroscopy table, then the address to choose if the start address + index –1 (index =10 in this case).

Table	Start Address	End Address
Write Simulated Data Parameter	0x4024265b	0x40242673
Write Detector Constants in Spectroscopy	0x40242673	0x4024268b
Write Detector Constants in Photometry	0x4024268c	0x402476a4

Table 25. Important Physical addresses in the Actual SW Version

Appendix A: SPU HLSW Architecture

```
!*****************************************************************************
! File name: spuhlsw.ach
! Version.Revision: HPSPU-HLSW-OB-V10R1
!
! Purpose: Mapping of the HLSW
!
! Public Functions: None
!
! Private Functions: None
!
! Description: This File describes the Segment Map of the SPU HLSW.
!
! Creation date & author: 10.06.2002 by Ahmed Nabil Belbachir
! Version, Update date & author: V10R1 01.06.2004 by Ahmed Nabil Belbachir
!*****************************************************************************
.system spu;
! Begin of Program Memory Segments
!===========================================================================
! This Segment Contains the SW Interrupt Table
.segment /pm /ram  /begin=0x00000100   /end=0x000002FF   seg_rth;

! This Segment Contains the C Initialization Code
.segment /pm /ram  /begin=0x00000300   /end=0x000009FF   seg_init;

! This Segment Contains the Program Code
.segment /pm /ram  /begin=0x00000A00   /end=0x0000AFFF   seg_pmco;

! This Segment Contains the Variables declared in Program Memory
.segment /pm /ram  /begin=0x0000B000   /end=0x0001FFFF   seg_pmda;

! This Segment is used In Buffer Transmission Mode
.segment /pm /ram  /begin=0x00020000   /end=0x0007FFFF   seg_pbuf;

! End of Program Memory Segments
!===========================================================================
```

```
! Begin of Data Memory Segments
!=======================================================================

! This Segment contains the SUSW Boot-up Results (ASW should not write in this
Memory)
.segment /dm /ram  /begin=0x00000000   /end=0x0000009F   seg_susw;

! This Segment contains the Circular Buffer for DEC/MEC Header
.segment /dm /ram  /begin=0x000000A0   /end=0x000034AF   seg_dmch;

! Segment not used
.segment /dm /ram  /begin=0x000034B0   /end=0x000034FF   seg_rxtx;

! This Segment contains the Circular Buffer for Science Data
.segment /dm /ram  /begin=0x00003500   /end=0x0003F57F   seg_scda;

! This Segment is used to store Temporary Results
.segment /dm /ram  /begin=0x0003F580   /end=0x00071FFF   seg_tpda;

! This Segment is actually not used by HLSW
.segment /dm /ram  /begin=0x00072000   /end=0x0007FFFF /cheap seg_heap;

! This Segment is used for Static Data Allocation
.segment /dm /ram  /begin=0x40200000   /end=0x40256FFF   seg_dmda;

! This Segment is used for the Stack Space
.segment /dm /ram  /begin=0x40257000   /end=0x4026FFFF /cstack seg_stak;

! This Segment is used for the Output TM Buffering
.segment /dm /ram  /begin=0x40270000   /end=0x4027FDFF seg_tmcb;

! This Segment is used for the memory scrubbing results on DRAM
.segment /dm /ram  /begin=0x4027FE00   /end=0x4027FEFF seg_scrd;

! This Segment is used for the memory scrubbing results on PRAM
.segment /dm /ram  /begin=0x4027FF00   /end=0x4027FFFF seg_scrp;
```

```
! End of Data Memory Segments
!=========================================================================

! Begin of COMI Segments. Only used in Case of the Simulator
!=========================================================================

.segment /dm /ram  /begin=0x40380000   /end=0x40381FFF   seg_dprm;

! End of COMI Segments
!=========================================================================

!Begin Configuration used by HLSW
!=========================================================================
! Bank PM0: PROM
.bank /pm0 /wtstates=0 /wtmode=external /begin=0x000000;

! Bank PM1: Program RAM, EEPROM, PMPSC registers.
.bank /pm1 /wtstates=0 /wtmode=external /begin=0x000000;

! Bank DM0: Data RAM
.bank /dm0 /wtstates=0 /wtmode=external /begin=0x00000000;

! Bank DM1: SMCS registers
.bank /dm1 /wtstates=0 /wtmode=external /begin=0x20000000;

! Bank DM3: Extended Data RAM, SMCS DPRAM, DMPSC registers
.bank /dm3 /wtstates=0 /wtmode=external /begin=0x40200000;

!End of Configuration used by HLSW
!=========================================================================

.endsys;
```

Appendix B: SPU HLSW Memory Map

Appendix B.1: HLSW Memory Map in SPU RAM

PRAM:

	Address
reserved length[2]: `256 words`	`0x00000000`
seg_rth Runtime Header Segment length: `512 words`	`0x00000100`
seg_init Initialisation Segment length: `1792 words`	`0x00000300`
seg_pmco Program Memory Code Segment length: `42496 words`	`0x00000A00`
seg_pmda Program Data Segment length: `86016 words`	`0x0000B000`

seg_pbuf
Program Buffer Segment
length: `393215 words`
	`0x00020000`
	
	`0x0007FFFF`

[2] The length is given in 6-Byte words.

DRAM:

	Address
seg_susw SUSW Boot-up Segment length[3]: `160 words`	`0x00000000`
seg_dmch Circular Buffer for DMCH Segment length: `3410 words`	`0x000000A0`
seg_rxtx Segment not used length: `80 words`	`0x000034B0`
seg_scda Circular Buffer for Science Data Segment length: `245888 words`	`0x00003500`
seg_tpda Temporary Data Segment length: `248448 words`	`0x0003F580`
seg_heap actually not used by HLSW length: `12288 words`	`0x0007D000` `0x0007FFFF`

extended DRAM:

	Address
seg_dmda Data Memory Data Segment Length[3]: `368640 words`	`0x40200000`
seg_stak Stack Segment Length: `143360 words`	`0x4024D000`
seg_tmcb Output TM Buffering (circular) length: `65024 words`	`0x40270000`
seg_scrd / seg_scrp Memory Check Results Storage (circular) length: `512 words`	`0x4027FE00` `0x4027FFFF`

[3] The length is given in 4-Byte words.

Appendix B.2: HLSW Memory Map in SPU EEPROM

Logical Address		Physical Address
0x00000000	reserved length[4]: 256 words	0x000E0000
0x00000100	**seg_rth** Runtime Header Segment length: 512 words	0x000E0100
0x00000300	**seg_init** Initialisation Segment length: 1792 words	0x000E0300
0x00000A00	**seg_pmco** Program Memory Code Segment length: 42496 words	0x000E0A00
0x0000B000	reserved length: 217087 words	0x000EB000
0x0003FFFF		0x0011FFFF

[4] The length is given in 6-Byte words.

Appendix C: Error messages and recovery procedures

C.1 General

- If the copy to EEPROM fails, the failure is reflected in the HK (TBD). Recovery Procedure: TBD

C.2 Command Acknowledgement

In the following table the error codes of the Negative Acknowledgement (NACK) are listed:

Error Code	Error Definition	Load	Dump	Check	Perform	Write	invalid	Recovery Procedure
0x71	invalid Command ID						●	Check Command ID is one of Load, Dump, Check, Perform or Write.
0x72	invalid Memory ID	●	●	●				Check if Memory ID is in the allowed memory area.
0x73	invalid Address	●	●	●				Check address and Memory ID.
0x74	Length not compatible to Parameter ID					●		Check Write command syntax.
0x75	invalid Checksum	●				●		Verify checksum.
0x76	invalid Parameter ID					●		Check if Parameter ID is one of Stop, Start, Test Mode, etc.
0x77	invalid Activity ID				●			Check Activity ID.
0x78	invalid Structure ID				●			Check Structure ID.
0x79	invalid SID Parameter				●	●		Check Structure ID parameter and Structure ID.
0x7B	invalid Checksum after re-reading	●				●		Send again; if problems persist, there is likely a HW problem.

Table 26. Error Codes and Recovery Procedures of Command Acknowledgement

C.3 HK

DEC/MEC Header Error (SPUS_DMC_ERROR / SPUL_DMC_ERROR): This field is set to a message code if an error is detected in the DEC/MEC header. See the list of errors in Table 19

Error Code	Description	Recovery Procedure
0x01	**Case Photometry:** DBID with value 1 or 5 has not been received to start a new buffer for data storage.	Send "Start"; PACS Safe Mode; Reconfigure DEC/MEC
0x02	**Case Photometry or Spectroscopy:** transition buffer due to a CMM change is ignored because its size is smaller than the minimum required granularity (see AD001).	Check DEC/MEC script; PACS Safe Mode; restart procedure
0x03	**Case Blue Photometry:** SPU SW did not receive 4	Check DEC/MEC script; PACS Safe Mode;

	consecutives sub-frames with DBID incrementing from 1 to 4.	restart procedure
0x04	**Case Photometry:** wrong SPUID (expected value 2 or 3).	Check DEC/MEC script; PACS Safe Mode; restart procedure
0x05	**Case Photometry:** DBID is out of the range [0,5].	Check DEC/MEC script; PACS Safe Mode; restart procedure
0x06	**Case Red Photometry:** SPUID (expected value 3) is not compatible with DBID (expected value 5).	Check DEC/MEC script; PACS Safe Mode; restart procedure
0x11	**Case Spectroscopy:** The frame with RRR = CRCRMP-1, has not been received to start a new buffer for data storage.	Start DEC/MEC?
0xAA	**Case Photometry or Spectroscopy:** wrong compression mode.	Check DPU command.
0xBB	Wrong command: Peak-up started for photometry data type	Check compression mode.
0xCC	Wrong command: Bolometer Background Canceling mode started for spectroscopy data type	Check compression mode.
0xDD	Data Incomplete. This happens when the CPU load is too high for seamless data processing.	Check detector constants. Is the subramp/submean length correct?
0xFF	Wrong observing mode (expected type values are 1 or 2)	Check DPU script and DEC/MEC.

Table 27. DMC Header Error Codes Descriptions and Recovery Procedures

Invalid LLC Parameter (SPUS_LLC_ERROR / SPUL_LLC_ERROR): This HK parameter reports an invalid write parameter for LLC. Default = 0.

Error Code	Description	Recovery Procedure
≠ 0x00	Signals an invalid write parameter for LLC.	Check SPU configuration script (write detector constants for spectroscopy/ photometry).

Table 28. Invalid LLC Parameter Description and Recovery Procedure

Memory Status Counters (SPUS_MEM_STATUS_CNTS / SPUL_MEM_STATUS_CNTS);

- LSB: EDAC memory check counter for Single Error Failure
- MSB: EDAC memory check counter for Double Error Failure

Error Code	Description	Recovery Procedure
≠ 0x00	LSB: Number of Single Error Failure MSB: Number of Double Error Failure	If memory errors persist, the HLSW has to be recompiled avoiding any corrupted memory area or the redundant unit shall be used.

Appendix D: Glossary

ASW .....Application SoftWare.

DEC/MEC .....Detector Controller/Mechanisms Controller.

DPU .....Digital Processing Unit.

EEPROM .....Electrically Erasable Programmable Read Only Memory.

HLSW .....High Level SoftWare.

ID .....IDentifier.

LLSW .....Low Level SoftWare.

LWL SPU .....Long WaveLength SPU. Also called Red SPU.

RAM .....Random Access Memory. It is

SPU .....Signal Processing Unit.

SUSW .....Start-Up SoftWare.

SWL SPU .....Short WaveLength SPU. Also called Blue SPU.

TC .....TeleCommand

TM .....TeleMetry

Appendix E: Reduction Algorithms

Here some insight on the most important reduction operations (fitting, averaging and rounding) is given. For a more detailed description please refer to AD001.

E.1 Ramp Fitting

In short, (sub)ramps are fit with a least squares algorithm, multiplied with the length of the (sub)ramp and truncated to integer. The slope of this fitting method is generated as follows:

$$slope = \frac{\sum x \sum y - n \sum xy}{\left(\sum x\right)^2 - n \sum x^2}$$

where

$\sum x$:= `sumX` $\quad \sum x^2$:= `Sum_of_X_Square` $\quad \left(\sum x\right)^2$:= `Sum_of_X_By_X`

$\sum xy$:= `sumXY` $\quad \sum y$:= `sumY`

are float values. The result is an integer value where the float result is roughly rounded and multiplied with the length of the (sub)ramp. Here is an example where the reset interval is 64 and the subramp length is 4:

Reset interval RRR = **64**

Subramp length SL = **4**

X	Y
0	12457
1	12509
2	12563
3	12616

`sumX` = **6.**

`Sum_of_X_Square` = **14.**

`Sum_of_X_By_X` = **36.**

`sumXY` = **75429.**

`sumY` = **50145.**

The numerator is:

`slope_z` = `sumX sumY – n sumXY` = 300870. – 301716. = **-846.**

The denominator is:

`slope_n` = `Sum_of_X_By_X – n Sum_of_X_Square` = 36. – 56. = **-20.**

The result is therefore:

`slope` = (`slope_z/slope_n`)`*SL±0.5` = **42.3*4+0.5** (float) => `slope` = **169** (integer)

E.2 Averaging and Rounding

Both, photometry and spectroscopy use the same improved averaging algorithm in version 13.95. This algorithm has support for additional rounding built-in. The following steps describe the “Roland” strategy that is used as default.

1. Given an averaging length of n samples/average, the first step is to calculate the inverse denominator d = 1.0 / (n * 2^r), where r is the number of bits to be rounded. Note that d is a float.
2. n samples are summed up and multiplied with d. The derived average is still a float.
3. In case the average is positive, 0.5 is subtracted. If it is less than zero, 0.5 is added. The alternative “Koryo” setting makes this step vice versa.
4. d is subtracted at random from the average. This ensures statistical correctness of cases where the fraction of the average is exactly 0.5.
5. The average is now cast (with truncation) to an integer.

☞ If the length of the dataset is not divisible by n, the remaining samples are treated with their own (correct) denominator.

☞ Note that the resulting average is right shifted by the number of rounding bits. This is automatically undone during decompression in PCSS.

☞ Due to the nature of the mathematical operation, the results from slope fitting and mean averaging have different units. The unit of the mean is still ADU, whereas the unit of the slope depends on the length of the subramp. An example: We have a ramp of 64 samples, which are uniformly increasing by 3. That dataset is fit with four 16-sample subslopes. The result will be 48 for each of the four subslopes, i.e. the internally derived slope (ADU/sample) multiplied with the number of samples of the subramp. So, in order to get back to ADU/sample, you have to divide by the subramp length (here: 48/16=3). If the desired unit is to have ADU/ramp, then you have to multiply by the number of subramps per ramp (here: 48*4=192).

Appendix F: Parameters for your scripts

The software version 13.95 has the standard modes that were/are used in FM_ILT (in version 13.8) plus enhanced functionality that is enabled with a few parameters. The two parameter tables that are of importance are WRT_DET_CST_PHOT and WRT_DET_CST_SPEC. The new parameters were chosen in a way that old scripts will continue to work with improved algorithms. For enabling new functionality such as rounding, decimation or sync delay, the scripts have to be adapted.

F.1 Photometry

This document has a detailed description of the WRT_DET_CST_PHOT parameters in the respective section. Three parameters are of special interest:

- NAVG, the number of samples to average. **Its default value is 0**, which means "do not override, take 4 in default compression mode and 8 in double compression mode". If NAVG has a different value, this overrides the compression mode setting, i.e. if you have "7" there, the SW will average 7 samples no matter whether you are using default mode or double mode or even half compression mode. All other modes (the ones with no reduction, like BTM or transparent mode) are not touched by this.
- Version 13.95 of the HLSW now has optional rounding of the averages in all compression modes and for all kinds of averaging lengths available. If RND is left at zero, no rounding is done. For high noise/high gain settings, a rounding of 2 bits is recommended to stay compatible with the datarate.
- RNDS is a new parameter that is best kept at the default value 0. It switches between three different averaging algorithms.
- DPRE and DPOST determine how many samples in the beginning and at the end of an averaging group are to be discarded. There should be no need to use that.
- Sync Delay is a new feature to allow for better timing of label sequences. A value of 0 leaves everything as it was, a value of 1 delays the start of a label sequence by 1 frame, etc. For allowed numbers look at the WRT_DET_CST_PHOT table in one of the preceding sections.

Here are some typical examples of parameter sets:

Scenario	PPF	PPP	TFP	TCP	GDFP	NAVG	RND	RNDS	SCM	LCAL	PCOD	DPRE	DPOST	SyncDel	SPR[10]
A	0	0	1000	100	1	0	0	0	1	0	3	0	0	0	0…
B	0	0	1000	100	1	0	2	0	1	0	3	0	0	1	0…
C	0	0	1000	100	1	8	1	0	1	0	3	0	3	1	0…

Scenario A: Probably from an old script. The software has no override of the number of samples to average and will therefore take 4 samples in default mode, 8 in double mode and 2 in half compression mode for averaging. No rounding is done, so this setting is a good one for a **low noise/low gain** setup.

Scenario B: There are two differences in this setup. First, a 2-bit rounding is activated here, probably for a **high gain/high noise** setup. Second, a sync delay of 1 frame is set.

Scenario C: Someone knows exactly what he or she is doing: average 5 and leave the next 3 out. This is done by setting the averaging length to 8 and the DPOST to 3. The remaining 5 samples will be averaged and rounded by one bit and also a sync delay of 1 frame is also used.

F.2 Spectroscopy

Again find the detailed description of the WRT_DET_CST_SPEC parameters in the respective section. In Spectroscopy the old parameters are still intact. But to really take advantage of the new capabilities, three new parameters are given below:

- RNDB, the number of bits to round. **Its default value is 0**, which means no rounding of course. This parameter has no effect if ramp fitting is performed.
- RNDS is the same as in photometry. Just live with the **default value 0**.
- FIXRC is a new feature to prevent the additional raw channels from rotating. Leave it at the default value 0 if the rotating is okay, or set to 1 to fix them at the initial value.
- DPRE and DPOST are also available in spectroscopy. They have a different meaning than in photometry, because they apply to a full ramp and not to a submean group.

Scn.	PPF	PPP	TFS	TCS	TGD	RSRR	GDFS	RFAL	SOAL	LCAL	PCOD	RNDB	RNDS	FIXRC	DPRE	DPOST	SPR[8]
A	0	0	1000	100	100	16	1	1	5	4	3	0	0	0	0	0	0…
B	0	0	1000	100	1	8	1	1	5	4	3	2	0	1	0	0	0…
C	0	0	1000	100	1	8	1	1	5	4	3	1	0	0	7	1	0…
D	0	0	1000	100	1	8	1	0	5	4	3	0	0	0	0	0	0…

Scenario A: Probably from an old script. 16 samples to average and nothing new. If RSRR are set to 8, then this is a good setting for the large capacitance.

Scenario B: 8 samples/submean with 2-bit rounding, probably for the smallest capacitance. Note that here the raw channel handbrake (FIXRC) is also activated.

Scenario C: Someone wants to discard the first 7 samples and the last one of a ramp to get 7 averages of 8 samples. Additional rounding is set to 1 bit.

Scenario D: Some setup far from default: 8-sample subramp fitting. Note that rounding and decimation are not supported here.

Appendix G: Index