HERSCHEL

SPIRE On-Board Software User Manual

Document Ref.: SPIRE-IFS-PRJ-001391

Draft: 0.5

Prepared by: Sergio Molinari

Distribution List:

RAL	K. King
	B. Swinyard
	S. "Duke" Sidher
IFSI	R. Cerulli
	R. Orfei
	J. Liù

1 Introduction

1.1 Purpose of the document

This document describes in detail the procedures to start-up and run the SPIRE OBS, the contents of the TC packets to be uplinked in order to perform the required function, and the contents of the TM packets that the OBS generates. This document does not duplicate the information provided in RD2, but rather represents its complement for all that is not therein specified.

1.2 Acronyms and Glossary

AVM	Avionic Model
BC	Bus Controller
BP	BreakPoint
BSW	DPU Boot Software
CDMS	Command and Data Management System
DM	Data Memory (DSP)
DPU	Digital Processing Unit
DSP	Digital Signal Processor
DTST	Dedicated Test Software Tools
EGSE	Electrical Ground Support Equipment
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESA	European Space Agency
HERSCHEL	Herschel Space Observatory
HK	Housekeeping
HW	Hardware
ICE	DSP In-Circuit Emulator
I/F	Interface
IFSI	Istituto di Fisica dello Spazio Interplanetario
NA	Not Applicable
OBS	On-Board Software
PM	Program Memory (DSP)
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
S/C	Spacecraft
S/S	Subsystem
SUT	Software Under Test
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TC	Telecommand
TM	Telemetry
VME	Virtual Machine Executable Code

1.3 Document List

1.3.1 Applicable Documents

Document Reference	Name	Number/version/date
AD1	SPIRE OBS User Requirements Document	
AD2	SPIRE OBS Software Specifications Document	
AD3	Packet Structure Interface Control Document (PSICD)	SCI-PT-ICD-7527 Issue 2.0
AD4	Herschel/Planck Instrument Data Rates	H-P-1-ASPI-TN-0204 Issue: 1

1.3.2 Reference Documents

Document Reference	Name	Number/version
RD1	DPU/ICU Spacecraft Interface Test Plan	
RD2	SPIRE Data ICD	
RD3	DRCU/DPU ICD	Sap-SPIRE-CCa-076-02 v0.7
RD4	Virtual Machine Compiler and Simulator	
RD5	MCU/SCU Command List	
RD6	DPU-BSW Software Requirement Document	DPU-SQ-CGS-001

2 OBS Initialisation

The OBS is resident in EEPROM. When switched on, the BSW is copied from PROM to PM. The details of the boot procedure can be found elsewhere (see RD6), here we simply note that if all the tests carried out are passed, the boot enters an infinite loop waiting for a command. Only in case of trouble (some memory area corrupted) an event (5,4) is reported.

Two kinds of commands can be sent at this point: a memory load (6,2) to upload a new image, or a trigger command (8,4). To start the application program the trigger command "Force boot" can be used: once sent, the application program is copied from EEPROM to PM and then started. At this point, if the DPU is connected to the CDMS simulator, HK packets will be received. This is the confirmation that the startup procedure has been successfully completed. If an event has been generated, as a rule of thumb, it is better to stop operations and to alert some DPU expert.

To upload a new image there are two possibilities: if the DPU has been switched on, the BSW is waiting for a command so that the user can send all the memory load commands. If the application software is running, the user has to send the "Jump_to_Boot" TC (8,4, 0xCA-0x08) in order to start again the BSW (after the application program starts, the BSW is still in PM since it occupies a memory area not used by the application software; only the interrupt table needs to be rewritten). After the memory load via TC (6,2) has been completed, the command "Load TC and boot" makes DPU copy the full image from DM to PM and then the application program is started. If large areas of DM are damaged so that there is not enough space to store the image

before copying it in PM, it is possible to upload a subset of TC. After any subset has been uploaded, the command to send is "Load TC and wait": when the BSW receives this command, this part of the image is copied in PM but the application program is not started. The BSW waits for the next subset and so on. When the last subset of memory packets is received, by sending the command "Load TC and boot" the DPU copies this last piece of code and then starts the execution of the application software. This command has not been tested so far and it should not be used.

The command "Force the boot" has been already explained. It is necessary to run the application program after a successful start-up. However, it can also be used after the BSW has generated an event (5,4). If, for instance, the BSW found corrupted memory in a zone not used by the application software, the user can in any case decides to force the boot of the application software. This command then, forces the DPU to copy the program from EEPROM to PM as no error had been found during the tests at start-up.

The C program **TCGen.c** provided by Gavazzi is available to translate the OBS image into a list of TC (6,2); at that point the "**ObswLoader**" procedure in SCOS2000 can be used to uplink the TCs to the DPU.

3 Instructions

3.1 Telecommand Verification

The generation of the telecommand verification packets TM (1,3) (Execution Start), TM (1,5) (Execution Progress) and TM (1,7) (Execution End) are controlled by the Ack bits in the TC sent to the DPU, as specified in AD3 (§3.1). The TM (1,1) (successful TC verification), TM (1,2) (unsuccessful TC verification) and TM (1,8) (Execution Failure) are issued by the OBS irespectively of the TC Ack bits. The actual dispatching fo these TM packets to the CDMS can be inhibited using service 14 (Packet Transmission Control) as specified in AD3.

Error codes contained in TM (1,2) are listed in AD3, while error codes in TM (1,8) are reported in RD2.

3.2 Housekeeping & Diagnostic Data Reporting

The OBS only generates HK packets TM (3,25). No Diagnostic packets are generated. The HK packet definition is stored in tables in the OBS. Four independent HK packets can be generated simultaneously, each with its own sampling interval.

HK Packet ID	Packet
0	Critical HK Packet
1	Nominal HK Packet. A subset of the parameters contained in this packet will be used for monitoring.
2	Free
3	Free

Table 3-1 List of allowed HK packets

The OBS does not perform any check on the DPU workload implied by the HK parameters collection. In particular, it should be remembered that the minimum time to issue a HK parameter request to the DRCU and receive the correspondent parameter is 2 milliseconds. This

means that nominally the cumulative number of DRCU parameters requested for the various HK packets should not exceed 500/sec to avoid losing data. In reality the number should be kept below that limit because the OBS is performing other tasks at the same time.

3.2.1 Situation at Start-Up

At OBS start-up two types of HK packets are generated by default: the critical HK packet and the nominal HK packet. Both packets are TM (3,25) and the header only differs for the SID (0x0300 and 0x0301 respectively). The two packets are issued every 2s and every 1s respectively. The definition of the two packets is loaded on OBS initialisation and complies with requirements contained in RD2.

3.2.2 Modifying the HK Packet Properties

3.2.2.1 Sampling Interval

The sampling interval of an HK packet can be modified via a TC (3,1), inserting the required interval in milliseconds in the proper TC field as specified in RD2. the new sampling interval is applied at the start of HK sampling cycle immediately following TC (3,1) reception. This means that if the current sampling interval of an HK packet is 10 seconds and a TC (3,1) with a 1 second sampling interval is received 2 seconds after the last HK packet has been issued, the 1-second HK packets will start to be sent after about 8 seconds from TC reception.

3.2.2.2 HK Packet Contents

The contents of the HK packets are defined by on-board tables that contain the list of DPU/DRCU 32-bits commands needed to get those parameters. The order in which the commands are stored in the HK definition tables defines the order in which the HK parameters are stored in the HK packets. To modify the contents of an HK packet, the first thing to do is then to uplink a new table (with its own ID number) containing the list of 32-bits commands needed to get the required HK parameters. The sequence of actions is then the following:

   a) Load a new HK definition Table. The mechanism to do this will be explained when dealing with Tables management.
   b) Stop HK acquisition using a TC (3,3) with the required HK Packet ID as specified in RD2.
   c) Restart HK acquisition using a TC (3,1) with the required HK Packet ID, the Table ID of the table uplinked in a) and the required sampling interval.

Warning: since the Nominal HK packet (ID 1) will be used for monitoring purposes, stopping HK Packet ID 1 will also stop the monitoring task. Besides, before redefining the table to be used for Nominal HK packet, it will be necessary to also redefine the table (TBD) which contains the definition of the monitored parameters.

Note: A TC (3,1) with a Table ID different from the one currently in use for that HK Packet ID must be preceeded by a TC (3,3), or a TM (1,8) with code 0x0307 (RD2) will be issued.

The list of commands for the DRCU is reported in RD3 and RD5. The list of commands to be uplinked to get DPU HK parameters is the following:

Command ID	Function	Param. bits
0x20010000	Get Observation ID	32
0x20020000	Get Building Block ID	32
0x10030000	Get Observing Mode	16
0x10040000	Get Observation Step	16
0x30050000	Get Time of Start HK0 parameters collection	48
0x301F0000	Get Time of Start HK1 parameters collection	48
0x30200000	Get Time of Start HK2 parameters collection	48
0x30210000	Get Time of Start HK3 parameters collection	48
0x30060000	Get time of last DRCU timer reset	48
0x10070000	Get ?	16
0x10080000	Get sequence number of last received TC	16
0x10090000	Get ?	16
0x100A0000	Get sequence number of last executed TC	16
0x100B0000	Get On-Target Flag	16
0x300C0000	Get time for OTF	48
0x100D0000	Get OTF error	16
0x100E0000	Get Telescope Scan Flag	16
0x300F0000	Get time of TSF	48
0x20100000	Get observation length in seconds	32
0x20110000	Get data rate of current observation in ?/s	32
0x30120000	Get Memory check info (TBD)	32
0x10130000	Get monitoring flags (TBD)	16
0x10140000	Get DCU flags (TBD)	16
0x10150000	Get SCU flags (TBD)	16
0x10160000	Get MCU flags (TBD)	16
0x04170000	Get DPU current	16
0x04180000	Get DPU 5V	16
0x04190000	Get DPU 15V	16
0x041A0000	Get DPU –15V	16
0x041B0000	Get DPU temperature	16
0x041C0000	Get DPU processor workload in parts per thousand	16

Table 3-2 Commands to get DPU HK parameters

3.3 Event Reporting

mettere qui gli ID necessari per interpretare gli eventi: poolID, taskID, FIFOID etc.

3.4 Table Management

All HK packet definitions and VM codes needed to perform the SPIRE observations are stored on-board as Tables. Each table is characterised by an ID and a length in 32-bits words. The absolute memory addresses of all on-board tables are managed by the OBS and are not

available to the user. The TCs to load and delete on-board tables are described in RD2. Here we describe how to use those TCs.

3.4.1 Table Load

The sequence to load a new table is the following:

a) Send a TC (8,4, 0x01-0x01) specifying the Table ID and the length in 32-bits words of the Table.
Warning: if the specified Table ID exists, the table is deleted. The only exception is if the table is in use (HK packet or VM code), in which case a TM (1,8) is issued with a TBD error code.
b) Send a TC (8,4, 0x01-0x03) containing the list of 32-bits words. Since the TC holds 16-bits words, each 32-bits words will have to be split in two, with the MSBs preceeding the LSBs. The number of the 32-bits words contained in the TC must not exceed the length specified in a) for that Table ID, or a TM (1,8) will be generated.

3.4.2 Table Update

To update an existing table it is sufficient to send a TC (8,4, 0x01-0x03) as specified in b) of 3.4.1.

3.4.3 Table Delete

To delete an existing table it is sufficient to send a TC (8,4, 0x01-0x01) specifying the Table ID and setting the length to 0.

Note: if the table is in use (HK packet or VM code) a TM (1,8) will be issued.

3.4.4 Table Defragmentation

Tables are stored in a dedicated DM area. After a while the continuous creation, update and deletion of tables may lead to an excessive memory fragmentation within that area. This may result in the inability to create new tables even when enough space is available if it is not contiguous. The OBS can defragment the DM either automatically when the free and contiguous space of the DM area dedicated to tables is less than 20%, or via a dedicated TC (8,4, TBD-TBD).

3.5 Virtual Machine Programs

VM programs are stored in tables in a dedicated DM area. RD4 describes how to write and compile a VM program using a GUI available under windows. The GUI is able to produce the executable VM code already organised in TC (8,4, 0x01-0x01) ready to be sent to the DPU.