SUBJECT: Operating the SPIRE Instrument

PREPARED BY: Sunil Sidher

DOCUMENT No: SPIRE-RAL-DOC-000768

ISSUE: 0.5 (DRAFT) **Date:** 31st May 2003

APPROVED BY:

Instrument Scientist (RAL) **Bruce Swinyard**	
DPU Engineer (IFSI) **Riccardo Cerulli-Irelli**	
DRCU Engineer (CEA) **Christophe Cara**	
MCU Engineer (LAM) **Didier Ferrand**	
ICC Development Manager (RAL) **Ken King**	
Instrument Development Manager (RAL) **Eric Sawyer**	

Distribution

Jean-Louis Auguerres	**CEA, Saclay**
Christophe Cara	**CEA, Saclay**
Riccardo Cerulli-Irelli	**IFSI**
Didier Ferrand	**LAM**
Hans-Gustav Floren	**SO**
Anna Di Giorgio	**IFSI**
Douglas Griffin	**RAL**
Peter Hargrave	**Cardiff**
Matt Griffin	**Cardiff**
Ken King	**RAL**
Tanya Lim	**RAL**
Sergio Molinari	**IFSI**
Goran Olofsson	**SO**
Renato Orfei	**IFSI**
David Smith	**RAL**
Bruce Swinyard	**RAL**

Change Record

ISSUE	DATE	
0.5	31st May 2003	Building Block sections split into two categories: - Basic operating modes - Observatory functions Added definitions of basic operating modes Updated definitions of the Photometer and Spectrometer Standby modes.

Table of Contents

Glossary

AOT	Astronomical Observation Template
BSM	Beam Steering Mirror
DCU	Detector Control Unit
DPU	Digital Processing Unit
DRCU	Detector Readout and Control Unit
FCU	Focal Control Unit
FTS	Fourier Transform Spectrometer
HK	Housekeeping
IF	Instrument Function
MCU	Mechanism Control Unit
MRS	MagnetoResistive Sensor
OBS	On-Board Software
PCAL	Internal calibrator for the Photometer
POF	Photometer Observatory Function
S/C	Spacecraft
SCAL	Internal calibrator for the FTS
SCU	Sub-system Control Unit
SMEC	Spectrometer Mechanism
SOF	Spectrometer Observatory Function
SPIRE	Spectral and Photometric Imaging Receiver

1. INTRODUCTION

The document "Operating Modes for the SPIRE Instrument" [AD-1] describes the instrument operating modes required to implement the types of observations for realising the science objectives of SPIRE.

The purpose of this document is to provide a description of the set of SPIRE instrument commands to be used in the Herschel ground segment and to be executed by the On-board Software (OBS) in order to carry out science observations. A typical science observation with SPIRE will consist of a number of Observatory Functions executed sequentially. An Observatory Function will itself consist of Instrument Functions, Data Configurations and S/C functions. Each component of an Observatory Function will be interpreted and expanded by the OBS into individual sub-system level commands before being sent to the appropriate DRCU sub-unit (viz. DCU, MCU and SCU).

This document also describes the types of science telemetry that needs to be generated in various operating modes. The actual structure and contents of the SPIRE telemetry packets are covered in [RD- 4].

1.1 Scope

The current document only discusses those instrument commands required for implementation of the Observatory Functions as described in [AD-1]. Future versions will describe the complete set of instrument operations, such as cooler recycle, memory checking and the switching between other modes. It has been found that the implementation of these commands may place requirements on the DRCU-DPU interface and so this preliminary version of the document is issued in order to ensure that these requirements are considered.

In this document the elaboration of Observatory Functions [AD-1] is confined to just the Instrument Functions and Data Configurations. The Spacecraft Functions necessary for the definition and execution of these Observatory Functions are identified and named, but not discussed here and it is assumed that they will be implemented as described in the Annexe "FIRST Pointing Modes" of [AD-3].

1.2 Assumptions

- For photometry observations all chop-throws will be performed in the Y direction. We assume that it will not be possible to change the orientation of the S/C arbitrarily to allow observers to specify their own chop positions on the sky (e.g. in terms of RA and Dec).

- At present there are no anomaly actions defined for the failure of an Observatory Function.

- It is further assumed that the execution of OBS commands will be independent of spacecraft triggers.

- The ISO observing strategy led to internal calibration measurements being routinely performed during each individual observation. A similar approach is also available for SPIRE observations but the Observatory Functions are designed in such a way that these calibrations can be performed outside the context of an observation. Consequently separate Observatory Functions exist for calibration.

2. Documents

2.1 Applicable Documents

AD-1	Operating Modes for the SPIRE instrument	SPIRE-RAL-DOC-000320	3.0	4th January 2002
AD-2	Herschel/Planck Packet Structure Interface Control Document	SCI-PT-ICD-07527	3 draft 6	24th January 2003
AD-3	Herschel/Planck Instrument Interface Document (IID Part A)	SCI-PT-IIDA-04624	2/0	31st July 2001

2.2 Reference Documents

RD-1	DRCU/DPU Interface Control Document	Sap-SPIRE-CCa-076-02	1.0	14th February 2003
RD-2	MCU/DCU [sic] Command List ICD and User Manual	LAM/ELE/SPI/01 1011	3.0	15th January 2003
RD-3	SPIRE Design Description	SPIRE-RAL-PRJ-000620	0.1	1st Sept 2000
RD-4	SPIRE Data Interface Control Document	SPIRE-RAL-PRJ-001078	1.0 (Draft 2)	15th Jan 2003
RD-5	Email from Christophe Cara on DRCU switch-on			28th May 2003
RD-6	Subsystem reactions for specification of the Instrument Simulator	SPIRE-RAL-NOT-00???	Issue 1	1st April 2003

3. OBSERVATORY FUNCTIONS

The detailed structure of an Observatory Function is illustrated schematically in Figure 1, with the flow of execution going from the top of the page to the bottom of the page.

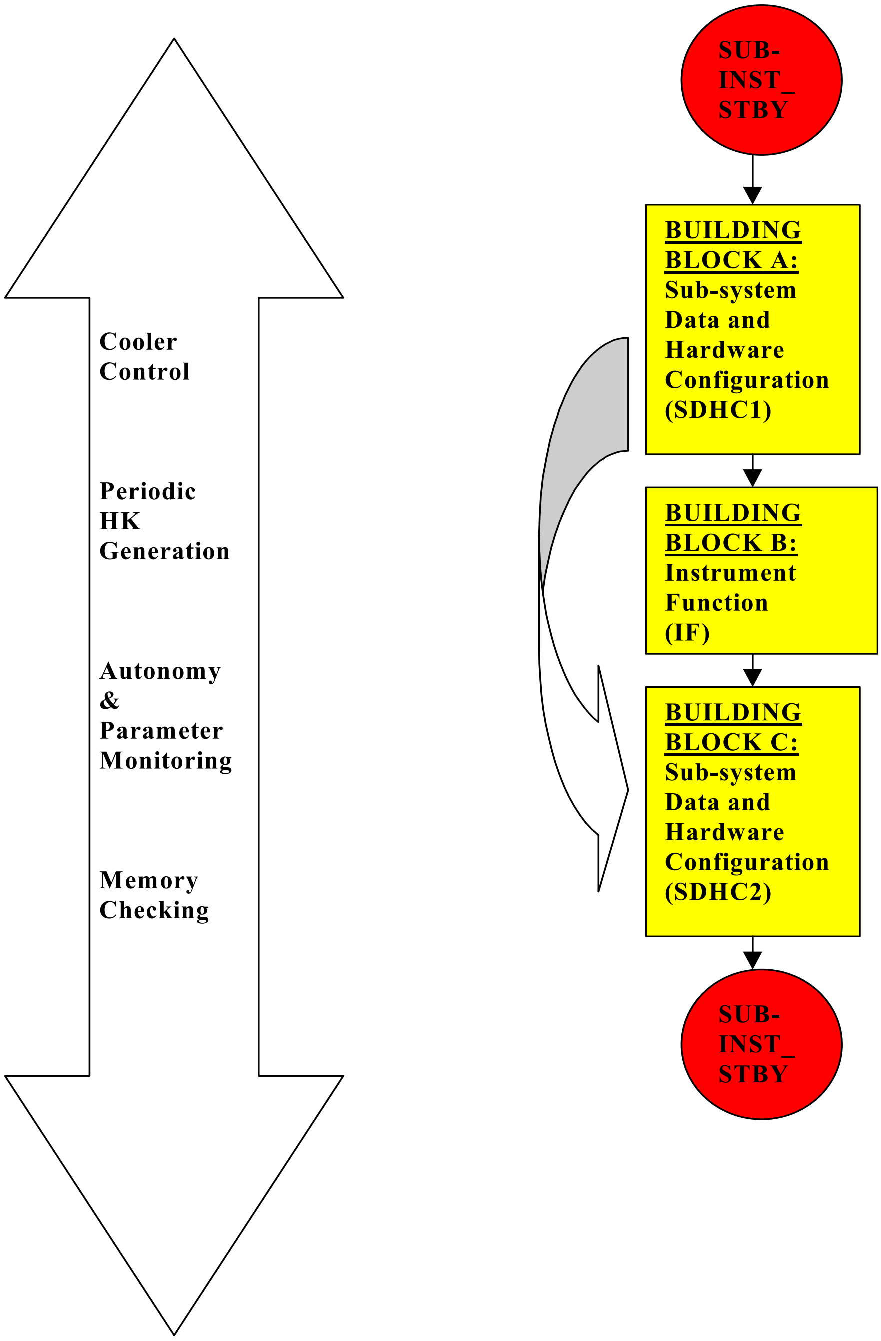

Figure 1: Structure of a SPIRE Observatory Function. The flow of execution is from top to bottom.

3.1 Nomenclature

- SPIRE Observatory Functions are designed to have all the functionality needed to perform complete astronomical observations (viz. AOTs) with the Photometer or the Spectrometer, including internal calibration. These Observatory Functions will consist of a sequence of Building Blocks, which can in principle comprise of both S/C and instrument commands. A BB is the key component of an Observatory Function like Photometer chopping, FTS scanning or internal calibration. In the Herschel ground segment each Observatory Function will have an Observation Identifier (ObsId) associated it.

- Each rectangular block in the diagram represents a BB. In the ground segment each BB has a BB Identifier (BBId) associated with it which simplifies the task of relating the down-linked telemetry with up-linked telecommands.

- Separate Observatory Functions are also available for performing internal calibration outside the context of an astronomical observation.

- Each BB includes sending commands to the DRCU sub-units, i.e. DCU, MCU and SCU.
  The sub-unit commands listed here have been take from [RD-1] where possible. These commands are prefixed with the name of the intended recipient of the DRCU sub-unit, e.g. DCU: SetPhotoBiasFreq.

- Parameters which the DPU has to set up are prefixed with the letters 'DPU'. DPU:SampleCount, for example, could be used by the OBS to tag sample numbers to detector data as received from the DCU.

- The BBs referring to S/C functions such as nodding, raster mapping and line scanning could span one or more instrument building blocks or may even last the duration of an Observatory Function.

3.2 Structure of an Observatory Function

A brief explanation of all the constituent steps of an Observatory Function, as shown in Figure 1, is now given.

3.2.1 SUB-INST_STBY mode

This is the assumed standby state of the instrument at the start of an Observatory Function. For observations with the Photometer SPIRE will start and end in the PHOT STBY mode whilst for Spectrometer observations it will start and end in the SPEC_STBY mode. The instrument will switch from REDY to either PHOT STBY or SPEC STBY *before* the start of an Observatory Function. In order to maximise observation efficiency and to minimise unnecessary switching of instrument states it is desirable to group together observations according to sub-instrument in use, i.e. Photometer or Spectrometer.

3.2.2 Building Block Definitions

As shown in Figure 1, and as explained below, each Observatory Function could potentially consist of three different types of Building Blocks for each type of instrument operation, i.e. BBs of types A, B and C. If an observation was to consist of a Photometer internal calibration called PCAL and Photometer chopping operation called CHOP then the Observatory Function could comprise the BBs shown in Figure 2.

PCAL A	PCAL B	PCAL C	CHOP A	CHOP B	CHOP C

Figure 2: An example Observatory Function showing the component Building Blocks

BBs of type B will always exist whereas types A and C will depend on the kind of operation to be performed with the instrument. All the SPIRE BB interface definitions and implementation commands are described in Section 7.

3.2.2.1 Building Block A: Sub-system Data and Hardware Configuration (SDHC1)

This BB is designed to

- select the housekeeping parameters to be sampled and their sampling rates
- configure the sub-systems for the Instrument Function to be executed
- switch on power to the mechanisms or sub-systems needed for the execution of the Instrument Function

For example, we may need to set up the instrument for a Photometer internal calibration measurement or for chop operations.

3.2.2.2 Building Block B: Instrument Function (IF)

This BB covers the instrument operations that have to be performed to carry out the actual scientific measurement part of the Observatory Function, including any internal calibration. A Photometer Observatory Function may consist of PCAL for performing a Photometer internal calibration measurement and CHOP for Photometer chop operations.

3.2.2.3 Building Block C: Sub-system Data and Hardware Configuration (SDHC2)

Sets up the data and hardware configuration of the instrument prior to the switching off of some mechanisms. The sampling rates of the detectors and the position of a mechanism can be selected here. This BB involves configuring the instrument for the end of an Observatory Function. The switching-off of power to the appropriate mechanisms also happens here.

3.2.3 SUB-INST_STBY

The instrument ends up in the sub-instrument STBY mode of the Observatory Function, i.e. PHOT STBY or SPEC STBY.

3.2.4 Other Tasks

During the execution of an observatory function there will be a number of other tasks which will be active throughout. These are also shown in Figure 1 and include:

- Active cooler control
- Periodic HK generation
- Monitoring of parameters
- Memory checking

4. MODE TRANSITIONS

Figure 3 shows a diagram of the SPIRE operating modes (same as Figure 3-1 in [AD-1]) and the various procedures for switching between these modes. As stated earlier the instrument will switch from the Ready mode (REDY) to Photometer standby (PHOT STBY) or Spectrometer standby (SPEC STBY) mode before the execution of Observatory Functions. The generic Observe mode (OBSV) in this figure corresponds to any one of the Observatory Functions to be discussed in Sections 5 and 6.

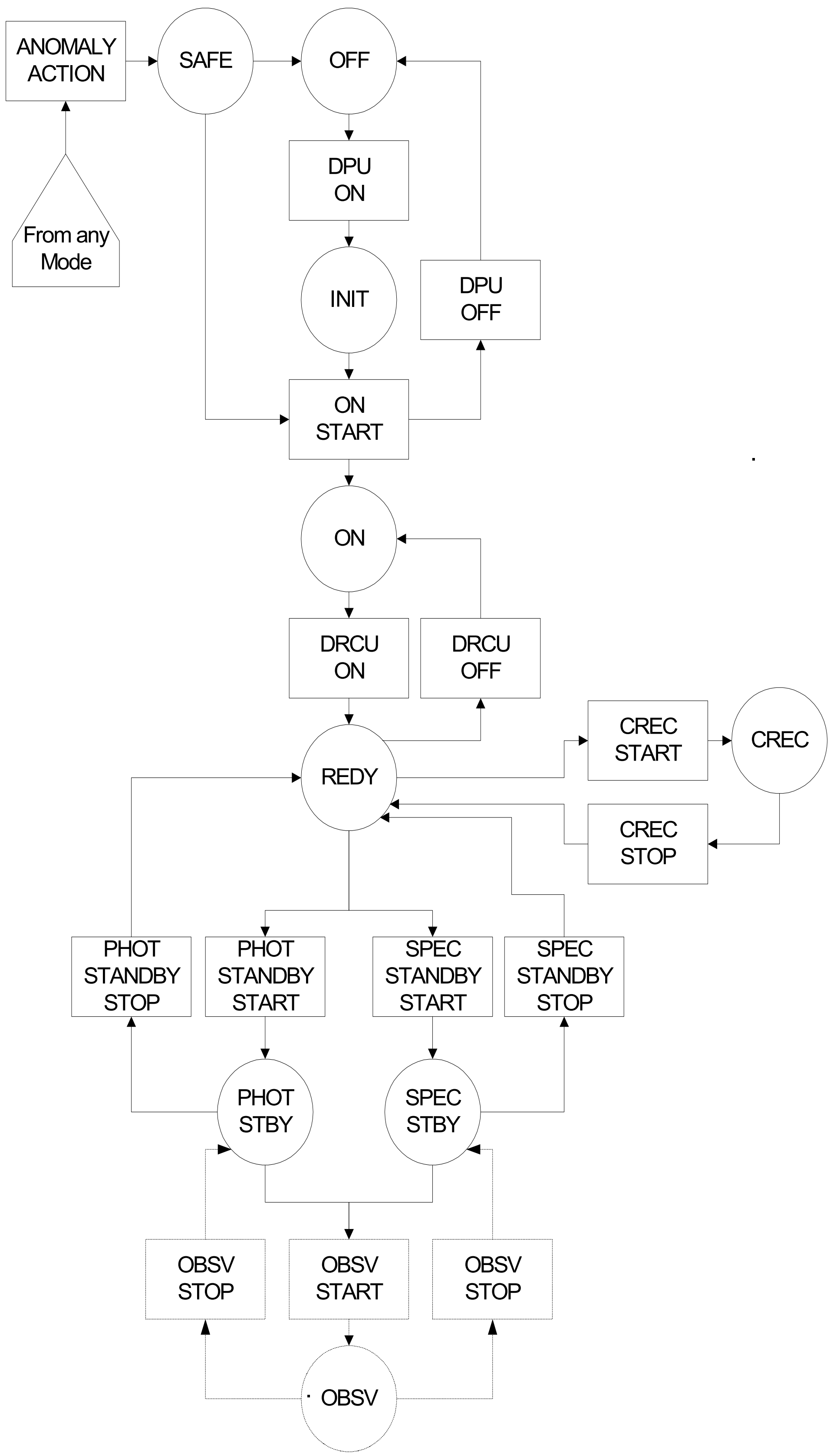

Figure 3: Logical transition flow between SPIRE operating modes. This diagram is taken from [AD-1].

5. PHOTOMETER OPERATIONS

5.1 POF1: Chop Without Jiggling

This is the simplest of the Photometer Observatory Functions. Its structure is expected to become a key component of other POFs, to be discussed later in this document. The high level operations to be performed in POF1 are as follows:

1) **TPOINT:** Point the telescope at the astronomical source
2) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
3) **PCAL:** Internally calibrate the Photometer *if necessary*
4) **CHOP:** Perform a series of chop cycles on the source while taking BDA data
5) **TNOD:** Nod the telescope and wait to settle
6) Repeat steps 4-5 as necessary
7) **PCAL:** Internally calibrate the Photometer *if necessary*
8) Repeat steps 4-7 as required
9) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

where **TPOINT, PHOT_DAQC, PCAL**, etc are the BBs in this POF. This scenario assumes that the BSM does not have to be switched off for PCAL operations (RD-3) and that Photometer calibration can be performed in the context of an astronomical observation.

- The OBS has to keep track of the number of chop cycles and the number of nod cycles.
- It is expected that the BDA data samples will be labelled with the sample number by the DCU.
- How do we select parameters to be sampled from SCU? Can they also be labelled with the sample number within the frame?

5.2 POF2: Seven-Point Jiggle Map

POF2 is intended for observing isolated compact sources with inaccurate co-ordinates or if the telescope pointing proves to be unreliable. In principle this POF should simply consist of POF1 being executed at each of the seven jiggle positions. It is envisaged that the jiggle sequence will be repeated a number of times to build up the required integration time. This is preferable to just performing a large number of chop cycles at each jiggle position and only visiting each jiggle position once. The order in which the jiggle positions are to be visited is TBD. (One possibility being discussed is that the BSM will return to the central jiggle position after visiting each of the other six jiggle positions). The parameters for the map will be read from a jiggle table stored in the OBS. The general sequence of steps is as follows:

1) **TPOINT:** Point the telescope at the astronomical source
2) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
3) **PCAL:** Internally calibrate the Photometer *if necessary*
4) Perform seven-point jiggle map
   - **JIGGLE:** Move the BSM to ith jiggle position (initially i=1)
   - **CHOP:** Perform a series of chop cycles on the source (as in POF1)
   - Increment jiggle position number
   - Repeat these steps in 4) for all seven jiggle positions
5) **PCAL:** Internally calibrate the Photometer *if necessary*
6) Repeat steps 4-5 as many times as the jiggle sequence is to be repeated
7) **TNOD:** Nod the telescope and wait to settle
8) **PCAL:** Internally calibrate the Photometer *if necessary*
9) Repeat steps 4-8 as many times as necessary
10) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

- It is assumed that the number of chop cycles will be the same at each jiggle position.
- The OBS has to keep track of the jiggle position number, the jiggle cycle sequence number, the chop cycle number (at each jiggle position) and the number of nod cycles (at each jiggle cycle sequence).

- It will probably not be necessary for the Photometer internal calibrator to be flashed for each jiggle position because [AD-1] states that the Observatory Function for PCAL (see POF8 below) will be executed at an interval of an hour or more. The above sequence allows the option of operating PCAL more flexibly if desired.

5.3 POF3: n-Point Jiggle Map

It is assumed that the sequence of steps in this POF is the same as for POF2.

1) **TPOINT:** Point the telescope at the astronomical source
2) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
3) **PCAL:** Internally calibrate the Photometer *if necessary*
4) Perform n-point jiggle map
   - **JIGGLE:** Move the BSM to i^{th} jiggle position (initially i=1)
   - **CHOP:** Perform a series of chop cycles on the source (as in POF1)
   - Increment jiggle position number
   - Repeat these steps in 4) for all n jiggle positions
5) **PCAL:** Internally calibrate the Photometer if necessary
6) Repeat steps 4-5 as many times as the jiggle sequence is to be repeated
7) **TNOD:** Nod the telescope and wait to settle
8) **PCAL:** Internally calibrate the Photometer if necessary
9) Repeat steps 4-8 as many times as necessary
10) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

- We assume that the number of chop cycles will be the same at each jiggle position.
- The OBS has to keep track of the jiggle position number, the jiggle cycle sequence number, the chop cycle number (at each jiggle position) and the number of nod cycles (at each jiggle cycle sequence).
- It will probably not be necessary for the Photometer internal calibrator to be flashed for each jiggle position because [AD-1] states that the Observatory Function for PCAL (see POF8 below) will be executed at an interval of an hour or more. The above sequence allows the option of operating PCAL more flexibly if desired.

5.4 POF4: Raster Map

A raster map is simply a rectangular grid of several n-Point Jiggle Maps (POF3) interleaved with Photometer internal calibrator flashes. The scenario set out below assumes that each raster point in the map may be visited more than once by simply repeating **TRASTER**, the telescope raster BB as specified in [AD-3]. The BSM is used to chop in the Y direction, which may not necessarily be perpendicular to the lines in the raster map.

1) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
2) **TRASTER:** Perform a raster map with the telescope
3) For each raster point in the map perform the following steps
   a) **DPU: WAIT:** Wait for the telescope to reach raster point
   b) **PCAL:** Internally calibrate the Photometer *if necessary*
   c) **JIGGLE:** Move the BSM to i^{th} jiggle position (initially i=1)
   d) **CHOP:** Perform a series of chop cycles on the source (as in POF1)
   e) Increment jiggle position number
   f) Repeat steps c) to d) for all n jiggle positions
   g) Repeat steps c) to f) as many times as the jiggle sequence is to be repeated at each raster point
   h) **TNOD:** Nod the telescope and wait to settle
   i) **PCAL:** Internally calibrate the Photometer if necessary
   j) Repeat steps c) to g) at the nod position as necessary
4) Repeat steps 2) and 3) if necessary
5) **PCAL:** Internally calibrate the Photometer if necessary
6) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

- For each raster point the OBS has to keep track of the jiggle position number, the jiggle cycle sequence number, the chop cycle number (at each jiggle position) and the number of nod cycles (at each jiggle cycle sequence).

5.5 POF5: Scan Map Without Chopping

A scan map without chopping is a continuous map of a region of the sky over a number of lines on the sky interleaved with Photometer internal calibrator flashes. No spacecraft Nodding is performed. In the following scenario it is assumed that scan map may be repeated a number of times.

1) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
2) **TSCAN:** Perform line scans on the sky using the telescope
3) **PCAL:** Internally calibrate the Photometer *if necessary*
4) **PSCAN:** Ask the DCU to collect science data while the telescope scans along a line on the sky
5) Increment line scan number after each line scan
6) **DPU: WAIT:** Wait for TBS time until the telescope is on the next line
7) Repeat steps 4-6 for all lines
8) **PCAL:** Internally calibrate the Photometer *if necessary*
9) Repeat steps 2-8 for as many times as necessary for the line scan map.
10) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

- The OBS has to keep track of the line scan count.
- The BSM may need to be commanded to the nominal "rest" position even for this POF because there needs to be active control over its position in case it starts to drift.
- It is assumed that the Photometer internal calibrator could be flashed at the start of the observation and then after the telescope has visited all the lines of the line scan at least once. [AD-1] states that POF8 will be executed at an interval of an hour or more.

5.6 POF6: Scan Map With Chopping

A scan map with chopping (for high 1/f noise) is a continuous map of a region of the sky over a number of lines on the sky interleaved with Photometer internal calibrator flashes. This POF is similar to POF1. The BSM is used to chop in the Y direction, which may not necessarily be in a direction perpendicular to the lines in the scan map.

1) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration
2) **TSCAN:** Perform line scans on the sky using the telescope
3) **PCAL:** Internally calibrate the Photometer *if necessary*
4) **CHOP:** Perform a series of chop cycles on the source for the duration of scan along line l
5) Increment line scan number
6) **DPU: WAIT:** Wait for TBS time until the telescope is on the next line
7) Repeat steps 4-6 for all lines
8) **PCAL:** Internally calibrate the Photometer *if necessary*
9) Repeat steps 2-8 for as many times as necessary for the line scan map.
10) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

- The OBS has to keep track of the line scan count and the chop cycle count.
- It is assumed that the Photometer internal calibrator could be flashed at the start of the observation and then after the telescope has visited all the lines of the line scan at least once. [AD-1] states that POF8 will be executed at an interval of an hour or more.

5.7 POF7: Photometer Peak-Up

TBW

5.8 POF8: Photometer Calibrate

A Photometer calibration takes place with the power to the BSM switched on but with the BSM not operating (i.e. the BSM is not chopping or jiggling) so that the only modulation of the detectors is as a consequence of the Photometer Calibrator (PCAL). It is assumed that the instrument will already be in PHOT_STBY mode with the Photometer detector arrays switched on.

1) **PHOT_DAQC:** Set the Photometer Data Acquisition configuration

2) **PCAL_ON:** Switch on PCAL subsystem
3) **DPU: WAIT:** Wait some TBS time
4) Reset counter for PCAL sequence
5) **PCAL:** Perform PCAL excitation sequence
6) **DPU: WAIT:** Wait some TBS time
7) Increment counter for PCAL sequence
8) Repeat steps 5-7 as necessary
9) **PCAL_OFF:** Switch off PCAL subsystem
10) **DPU: WAIT:** Wait TBS time before resuming operations
11) **PHOT_SDHC_RESET:** Reset the Photometer Data and Hardware Configuration

6. SPECTROMETER OPERATIONS

6.1 SOF1: Spectrum of Point Source (Continuous Scan)

It is assumed that the instrument will be in SPEC_STBY mode initially with the Spectrometer BDAs , SMEC and SCAL switched on.

1. **TPOINT:** Point the telescope at the astronomical source
2. **DPU: WAIT:** Wait for some TBS time to allow the SMEC to stabilise.
3. **SPEC_DAQC:** Set the Spectrometer Data Acquisition configuration (see Section 7.9)
4. **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
5. Initialise FTS scan counter
6. **SPEC_SCAN:** Perform an FTS scan
7. Increment FTS scan counter
8. Repeat steps 6-7 as required
9. **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
10. **SPEC_HDC_RESET:** Reset the Spectrometer Data and Hardware Configuration

6.2 SOF2: Fully Sampled Spectral Map (Continuous Scan)

It is assumed that the instrument will be in SPEC_STBY mode initially with the Spectrometer BDAs and SCAL switched on.

1. **TPOINT:** Point the telescope at the astronomical source
2. **DPU: WAIT:** Wait for some TBS time to allow the SMEC to stabilise.
3. **SPEC_DAQC:** Set the Spectrometer Data Acquisition configuration (see Section 7.9)
4. **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
5. Make a fully sampled spectral map
   a) Initialise the jiggle position counter
   b) **JIGGLE:** Move the BSM to i^{th} jiggle position (Initially i=1)
   c) Initialise FTS scan counter
   d) **SPEC_SCAN:** Perform an FTS scan
   e) Increment FTS scan counter
   f) Repeat steps d) and e) as many times as required
   g) Repeat steps b) to f) for the n jiggle positions
6. Repeat all the steps in 5) for as many jiggle cycles as necessary
7. **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
8. **SPEC_HDC_RESET:** Reset the Spectrometer Data and Hardware Configuration

6.3 SOF3: Spectrum of Point Source (Step-and-Integrate)

This Observatory Function is similar to SOF1, except that here the FTS mirror is stepped to a given position and the BSM used to perform a number of chop cycles between two positions on the sky. This process is repeated for all FTS mirror positions to build up a complete scan. The use of SCAL is not necessary for this Observatory Function.

1) **TPOINT:** Point the telescope at the astronomical source
2) **SPEC_DAQC:** Set the Spectrometer Data Acquisition configuration (see Section 7.9)
3) **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
4) Perform a spectral scan in step-and-integrate mode:
   a) Initialise FTS scan counter
   b) **SPEC_STEP:** Step the FTS mirror to required position
   c) **CHOP:** Perform a series of chop cycles from given FTS mirror position
   d) Repeat steps b) and c) until end of FTS scan is reached
   e) Increment FTS scan counter
   f) Repeat steps b) to e) for the required number of FTS scans
5) **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
6) **SPEC_HDC_RESET:** Reset the Spectrometer Data and Hardware Configuration

6.4 SOF4: Fully Sampled Spectral Map (Step-and-Integrate)

This Observatory Function is essentially SOF3 performed at each of the jiggle positions of an n-point jiggle map.

1) **TPOINT:** Point the telescope at the astronomical source
2) **SPEC_DAQC:** Set the Spectrometer Data Acquisition configuration (see Section 7.9)
3) **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
4) Perform a fully sampled spectral map in step-and-integrate mode:
   a) Initialise jiggle position counter
   b) **JIGGLE:** Move the BSM to the i^{th} jiggle position (initially i=1)
   c) Perform a spectral scan in step-and-integrate mode:
      I) Initialise FTS scan counter
      II) **SPEC_STEP:** Step the FTS mirror to required position
      III) **CHOP:** Perform a series of chop cycles from given FTS mirror position
      IV) Repeat above steps II) and III) until end of FTS scan is reached
      V) Increment FTS scan counter
      VI) Repeat steps II) to V) for required number of FTS scans at each jiggle position
      VII) Repeat all steps in b) and c) for all n jiggle positions
5) Repeat all of the steps in 4) for the required number of jiggle cycles
6) **SPEC_PCAL:** Internally calibrate the Spectrometer using the PCAL source *as necessary*
7) **SPEC_HDC_RESET:** Reset the Spectrometer Data and Hardware Configuration

7. BUILDING BLOCKS FOR THE SPIRE OPERATING MODES

7.1 DPU_ON: Switch on DPU (OFF to INIT)

Switch on the DPU to go from OFF mode to INIT mode.

7.1.1 Interface

TBW

7.1.2 Implementation

Command	Description	
CDMS: DPU_SWITCH_ON	The CDMS switches on the DPU by powering the relevant 28V line	
Parameter	***Description***	***Value***

Command	Description
DPU: FORCE_BOOT	Boot the DPU from the EEPROM

Parameter	*Description*	*Value*
FUNCTION ID	*Function Identifier*	*0x0070*
ACTIVITY ID	*Activity Identifier*	*0x0003*

7.2 ON_START: Switch from INIT to ON mode

In this mode any RAM patches can be loaded into the DPU from the CDMS. This procedure could take several minutes.

7.2.1 Interface

TBW

7.2.2 Implementation

Command	**Description**	
DPU: LOAD_MEMORY	Load OBS using a series of such commands	
Parameter	***Description***	***Value***
MEMORY ID	*Memory Identifier*	
STARTADDR	*Start Address*	
NSAU	*Number of single (16-bit) addressable units*	
DATA	*Contents of the NSAU memory locations*	

7.3 DRCU_ON: Switch on the DRCU (ON to REDY)

Switches on the DRCU to go from ON to REDY mode.

7.3.1 Interface

TBW

7.3.2 Implementation

Command	**Description**	
CDMS: DRCU_SWITCH_ON	The CDMS switches on the DRCU by powering the relevant 28V line	
Parameter	***Description***	***Value***

Command	**Description**	
SCU: SetTempOnOff	Set standard FPU temperature probes biases on/off	
Parameter	***Description***	***Value***
TempOnOff	*16 bits on/off word*	*TBD*

Command	**Description**	
SCU: SetSubKOnOff	Sets sub K temperature probe bias on/off	
Parameter	***Description***	***Value***
SubKOnOff	*Bit 0: 1- bias on, 0- bias off*	*On (Bit 0=1)*

Command	**Description**	
SCU: SetDrelOnOff	Switches on the MCU power relay	
Parameter	***Description***	***Value***
DrelOnOff	*Set MCU power relay on (Bit 2 set)*	*0x0004*

Command	**Description**

MCU: SetDownLoadConf	Copy from MCU PROM to program RAM and wait 10 seconds	
Parameter	***Description***	***Value***

Command	Description	
MCU: SetBootRam	Reset the MCU DSP on RAM	
Parameter	***Description***	***Value***
on	*Boot on RAM*	*on*

Assumptions:

- Issue SCU commands to enable temperature measurements (as per RD-5).

7.4 CREC_START: Start Cooler Recycling (REDY to CREC)

Switch from REDY to CREC mode.

7.4.1 Interface

Building Block	Description	
CREC_START	Start cooler recycling	
Parameters	***Description***	***Value***
EVHSHeatCur	*Evaporator Heat Switch heater current*	*0x0800 (TBD mA)*
InitialSPHeaterCur	*Initial setting for Sorption Pump Heater current*	*0x0708 (22 mA)*
HotSPHeaterCur	*Setting for Sorption Pump Heater current once pump is hot (> 40K)*	*0x0287 (7.9 mA)*
ZeroSPHeaterCur	*Zero setting for Sorption Pump Heater current*	*0x0000 (0.0 mA)*
SPHSHeatCur	*Sorption Pump Heat Switch heater current*	*0x0800?*
ZeroEVHSHeatCur	*Zero setting of Evaporator Heat Switch heater current*	*TBD (0 mA)*

7.4.2 Implementation

Command	Description	
SCU: SetEVHSHeatCur	Set current of Evaporator Heat Switch heater	
Parameter	***Description***	***Value***
EVHSHeatCur	*Evaporator Heat Switch heater current (12 bits)*	*0x0800?*

Command	Description	
SCU: SetSPHeaterCur	Set current of Sorption Pump Heater	
Parameter	***Description***	***Value***
SPHeaterCur	*Sorption Pump Heater current (12 bits)*	*0x0708? (for a heater setting of 200mW, corresponding to 22mA)*

Command	Description	
SCU: SetSPHeaterCur	Set current of Sorption Pump Heater once the pump gets hot (> 40K)	
Parameter	***Description***	***Value***
SPHeaterCur	*Sorption Pump Heater current (12 bits)*	*0x0287? (for a heater setting of 25mW, corresponding to 7.9mA)*

Command	Description	
SCU: SetSPHeaterCur	Set current of Sorption Pump Heater to zero after some time	
Parameter	***Description***	***Value***
SPHeaterCur	*Sorption Pump Heater current (12 bits)*	*0*

Command	Description	
SCU: SetSPHSHeatCur	Set current of Sorption Pump Heat Switch heater	
Parameter	***Description***	***Value***
SPHSHeatCur	*Sorption Pump Heat Switch heater current*	*0x0800?*

Command	Description	
SCU: SetEVHSHeatCur	Set current of Evaporator Heat Switch heater to zero, thereby opening the switch to allow the evaporator temperature to fall as the ^{3}He is pumped.	
Parameter	***Description***	***Value***
EVHSHeatCur	*Evaporator Heat Switch heater current (12 bits)*	*0x0000*

7.5 CREC_STOP: Stop Cooler Recycling (CREC to REDY)

7.5.1 Interface

TBW

7.5.2 Implementation

TBW

7.6 PHOT_STBY_START: Switch from REDY to PHOT_STBY mode

7.6.1 Interface

TBW

7.6.2 Implementation

As already described in Section 3, for all Photometer operations the instrument is expected to be in the PHOT_STBY mode. A possible implementation for switching from the REDY mode to PHOT_STBY mode is as follows:

Command	Description	
DCU: SetPhSWJfetPwr	Switch on PSW JFET drain voltage	
Parameter	***Description***	***Value***
PSW_JFET_1	*Bit 0 (0 – off, 1 –on)*	*1*
PSW_JFET_2	*Bit 1 (0 – off, 1 –on)*	*1*
PSW_JFET_3	*Bit 2 (0 – off, 1 –on)*	*1*
PSW_JFET_4	*Bit 3 (0 – off, 1 –on)*	*1*
PSW_JFET_5	*Bit 4 (0 – off, 1 –on)*	*1*
PSW_JFET_6	*Bit 5 (0 – off, 1 –on)*	*1*

Command	Description	
DCU: SetPhMLWJfetPwr	Switch on PMW and PLW JFET drain voltage	
Parameter	***Description***	***Value***
PMW_JFET_1	*Bit 0 (0 – off, 1 –on)*	*1*
PMW_JFET_2	*Bit 1 (0 – off, 1 –on)*	*1*

PMW_JFET_3	*Bit 2 (0 – off, 1 –on)*	*1*
PMW_JFET_4	*Bit 3 (0 – off, 1 –on)*	*1*
PLW_JFET_1	*Bit 4 (0 – off, 1 –on)*	*1*
PLW_JFET_2	*Bit 5 (0 – off, 1 –on)*	*1*
TC_JFET	*Bit 6 (0 – off, 1 –on)*	*1*

Command	**Description**	
DCU: SetPhotoBiasMode	Set the Photometer & TC bolometer sine bias mode	
Parameter	***Description***	***Value***
PhotoBiasMode	*00 – stop; 0x01 to 0xfe – discrete values; ff – run*	*0xef*

Command	**Description**	
DCU: SetPhotoHeaterBias	Start the heating the JFET modules by setting the PhotoHeaterBias	
Parameter	***Description***	***Value***
PhotoHeaterBias	*0 to 255*	*TBD*

Command	**Description**	
DCU: SetPhotoHeaterBias	Stop heating the JFET modules by setting the PhotoHeaterBias to 00	
Parameter	***Description***	***Value***
PhotoHeaterBias	*00*	*00*

Command	**Description**	
DCU: SetPhotoBiasFreq	Set the Photometer & T/C bolometer sine bias frequency division from the master clock	
Parameter	***Description***	***Value***
PhotoMclkDiv	*Master clock divider setting (64 to 511)*	*TBD*

Command	**Description**	
DCU: SetPhotoBiasAmplSW DCU: SetPhotoBiasAmplMW DCU: SetPhotoBiasAmplLW DCU: SetPhotoBiasAmplTC	These four commands set the sine bias amplitude for the Photometer arrays and T/C bolometers.	
Parameter	***Description***	***Value***
PhotoBiasAmplSW *PhotoBiasAmplMW* *PhotoBiasAmplLW* *PhotoBiasAmplTC*	*Bias amplitude DAC setting parameter (0 to 255)*	*TBD*

Command	**Description**	
DCU: SetPhotoDemodSW DCU: SetPhotoDemodMW DCU: SetPhotoDemodLW DCU: SetPhotoDemodTC	These four commands set the demodulation phase shift for the Photometer arrays and T/C bolometers.	
Parameter	***Description***	***Value***
PhaseShiftSW *PhaseShiftMW* *PhaseShiftLW* *PhaseShiftTC*	*Phase Shift*	*0 to 255*

Command	**Description**	
DCU: SetPhSWJfetVSS1 to SetPhSWJfetVSS6	Set the JFET source biasing voltages for PSW channels	
Parameter	***Description***	***Value***

PSW_VSS1 to PSW_VSS6	*Voltage setting for each JFET unit*	*0 to 255*

Command	**Description**	
DCU: SetPhMWJfetVSS1 to SetPhMWJfetVSS4	Set the JFET source biasing voltages for PMW channels	
Parameter	***Description***	***Value***
PMW_VSS1 to PMW_VSS4	*Voltage setting for each JFET unit*	*0 to 255*

Command	**Description**	
DCU: SetPhLWJfetVSS1 to SetPhLWJfetVSS2	Set the JFET source biasing voltages for PLW channels	
Parameter	***Description***	***Value***
PLW_VSS1 to PLW_VSS2	*Voltage setting for each JFET unit*	*0 to 255*

Command	**Description**	
DCU: SetTCJfetVSS1	Set the JFET source biasing voltages for TC channels	
Parameter	***Description***	***Value***
PSW_VSS1 to PSW_VSS6	*Voltage setting for each JFET unit*	*0 to 255*

Command	**Description**	
MCU: SetChopLoopMode	Open the chop loop	
Parameter	***Description***	***Value***
ChopLoopMode	*0 - chopper is not commanded.*	*0*

Command	**Description**	
MCU: SetCSensorPwr	Power up the magnetoresistive sensor (MRS)	
Parameter	***Description***	***Value***
CSensorPwr	*1 – sensor power on*	*1*

Command	**Description**	
MCU: SetBSMMove	Set the chopper axis to move free run (default value)	
Parameter	***Description***	***Value***
BSMMove	*Flag to indicate BSM movement to default chopper axis MRS position*	*0*

Command	**Description**	
MCU: SetChopTargetPosition	Set the chopper axis to desired home position (determined from a calibration table)	
Parameter	***Description***	***Value***
ChopTargetPosition	*MRS position*	*TBD*

Command	**Description**	
MCU: SetChopLoopMode	Close the chop loop at desired position	
Parameter	***Description***	***Value***
ChopLoopMode	*1 -* the chopper moves to home position	*1*

7.7 SPEC_STBY_START: Switch from REDY to SPEC_STBY mode

7.7.1 Interface

TBW

7.7.2 Implementation

For all Spectrometer operations the instrument is expected to be in the SPEC_STBY mode. A possible scenario for switching from the instrument REDY mode to the SPEC_STBY mode is as follows.

Command	**Description**	
DCU: SetSpSLWJfetPwr	Switch on Spectrometer JFET drain voltages	
Parameter	***Description***	***Value***
PSLW_JFET_1	*Bit 0 (0 – off, 1 –on)*	*1*
PSLW_JFET_2	*Bit 1 (0 – off, 1 –on)*	*1*
PSLW_JFET_3	*Bit 2 (0 – off, 1 –on)*	*1*

Command	**Description**	
DCU: SetSpectroBiasMode	Set the Spectrometer bolometer sine bias mode	
Parameter	***Description***	***Value***
SpectroBiasMode	*00 – stop; 0x01 to 0xfe – discrete values; ff – run*	*0xef*

Command	**Description**	
DCU: SetSpectroHeaterBias	Start the heating the JFET modules by setting the SpectroHeaterBias	
Parameter	***Description***	***Value***
SpectroHeaterBias	*0 to 255*	*TBD*

Command	**Description**	
DCU: SetSpectroHeaterBias	Stop heating the JFET modules by setting the SpectroHeaterBias to 00	
Parameter	***Description***	***Value***
SpectroHeaterBias	*00*	*00*

Command	**Description**	
DCU: SetSpectroBiasFreq	Set the Spectrometer sine bias frequency division from the master clock	
Parameter	***Description***	***Value***
SpectroMclkDiv	*Master clock divider setting (64 to 511)*	*TBD*

Command	**Description**	
DCU: SetSpectroBiasAmplSW DCU: SetSpectroBiasAmplLW	These two commands set the sine bias amplitude for the Spectrometer array bolometers.	
Parameter	***Description***	***Value***
SpectroBiasAmplSW *SpectroBiasAmplLW*	*Bias amplitude DAC setting parameter (0 to 255)*	*TBD*

Command	**Description**	
DCU: SetSpectroDemodSW DCU: SetSpectroDemodLW	These two commands set the demodulation phase shift for the Spectrometer array bolometers.	
Parameter	***Description***	***Value***
PhaseShiftSW *PhaseShiftLW*	*Phase Shift*	*0 to 255*

Command	Description	
DCU: SetPhSWJfetVSS1 to SetPhSWJfetVSS2	Set the JFET source biasing voltages for SSW channels	
Parameter	***Description***	***Value***
SSW_VSS1 to SSW_VSS2	*Voltage setting for each JFET unit*	*0 to 255*

Command	Description	
DCU: SetPhLWJfetVSS	Set the JFET source biasing voltages for SLW channels	
Parameter	***Description***	***Value***
SLW_VSS	*Voltage setting for each JFET unit*	*0 to 255*

Command	Description	
SCU: SetSCal4Bias	Set value of current applied to stimulate SCAL 4% source	
Parameter	***Description***	***Value***
SCal4CurSP	*Current value*	*0x068f (2.25 mA)*

Command	Description	
SCU: SetSCal2Bias	Set value of current applied to stimulate SCAL 2% source	
Parameter	***Description***	***Value***
SCal2CurSP	*Current value*	*0x068f (2.25mA)*

Command	Description	
MCU: SetChopLoopMode	Open the chop loop	
Parameter	***Description***	***Value***
ChopLoopMode	*0 - chopper is not commanded.*	*0*

Command	Description	
MCU: SetCSensorPwr	Power up the magnetoresistive sensor (MRS)	
Parameter	***Description***	***Value***
CSensorPwr	*1 – sensor power on*	*1*

Command	Description	
MCU: SetBSMMove	Set the chopper axis to move free run (default value)	
Parameter	***Description***	***Value***
BSMMove	*Flag to indicate BSM movement to default chopper axis MRS position*	*0*

Command	Description	
MCU: SetChopTargetPosition	Set the chopper axis to desired home position (determined from a calibration table)	
Parameter	***Description***	***Value***
ChopTargetPosition	*MRS position*	*TBD*

Command	Description	
MCU: SetChopLoopMode	Close the chop loop at desired position	
Parameter	***Description***	***Value***
ChopLoopMode	*1* - the chopper moves to home position	*1*

Command	Description	
MCU: SetSLoopMode	Open the SMEC optical encoder loop	
Parameter	***Description***	***Value***

SLoopMode	*0 - SMEC is not commanded.*	*0*

Command	**Description**	
MCU: SetEncoderPwr	Power up the optical encoder LED	
Parameter	***Description***	***Value***
EncoderPwr	*1 – optical encoder LED power on*	*1*

Command	**Description**	
MCU: SetSLVDTPwr	Power up the LVDT oscillator	
Parameter	***Description***	***Value***
SLVDTPwr	*1 – LVDT power on*	*1*

Command	**Description**	
MCU: SetSLoopMode	Close the SMEC optical encoder loop	
Parameter	***Description***	***Value***
SLoopMode	*1 – SMEC optical encoder is in closed loop*	*1*

Command	**Description**	
MCU: SetSTrajMode	Set the SMEC to home position	
Parameter	***Description***	***Value***
STrajMode	*0x0004 – start the SMEC initialisation procedure* *Wait for 20 seconds for completion. The DPU must send a polling command every second to the MCU during this period to check if any errors are occurring.*	*0x0004*

7.8 PHOT_DAQC: Set Photometer Data Acquisition and Configuration

7.8.1 Interface

TBW

7.8.2 Implementation

Command	**Description**	
DCU: SetPhotoBiasMode	Start the Photometer & TC bolometer sine bias generator mode	
Parameter	***Description***	***Value***
PhotoBiasMode	*00 – stop; 0x01 to 0xfe – discrete values; ff – run* *Wait xx ms for the system to become stable*	*0xff*

Command	**Description**	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run*	*0*

Command	**Description**	
DCU: SetPhotoMode	Set the LIA offsets for the Photometer & TC channels automatically	
Parameter	***Description***	***Value***

Mode	*Indicates that the LIA offsets are selected automatically. Not clear what these mode definitions mean.*	*10*

Command	Description	
MCU: SetTelemetryPacket12SampFreq	Set the sampling frequency for the BSM TM packet	
Parameter	***Description***	***Value***
TelemetryPacket12SampFreq	*BSM TM packet sampling frequency in units of scheduler cycles (1 cycle= 360 µs). 44 cycles (≡ 0x002c) correspond to a sampling frequency of 64 Hz*	*0x002c*

Command	Description	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*1*

Command	Description	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*0*

Command	Description	
DCU: SetPhotoMode	Set the LIA offsets for the Photometer & TC channels automatically	
Parameter	***Description***	***Value***
Mode	*Indicates that the LIA offsets are selected automatically. . Not clear what these mode definitions mean.*	*18*

Command	Description	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms for the offsets to be sent to the DPU*	*1*

Command	Description	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*0*

Command	Description	
DCU: SetDataMode	Set the DCU output data format between bolometer or 4-bit offset transmission and test pattern	
Parameter	***Description***	***Value***
DataMode	*00000 – Full Photometer (all 5 bits set to 0)*	*00000*

Command	Description
DCU: SetFramenber or SetFrameCount?	Set frames acquisition mode

Parameter	***Description***	***Value***
FrameCount or frame?	*0 – Continuous*	*0*

Command	**Description**	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*1*

- It is assumed here that the above sequence of acquisition and configuration commands can be executed before the DPU explicitly requests BDA data from the DCU. Until that happens there are no BDA data being generated.

7.9 SPEC_DAQC: Set Spectrometer Data Acquisition and Configuration

7.9.1 Interface

TBW

7.9.2 Implementation

Command	**Description**	
DCU: SetSpectroBiasMode	Start the Spectrometer sine bias generator mode	
Parameter	***Description***	***Value***
SpectroBiasMode	*00 – stop; 0x01 to 0xfe – discrete values; ff – run* *Wait xx ms for the system to become stable*	*0xff*

Command	**Description**	
MCU: SetTelemetryPacket10SampFreq	Set the sampling frequency for the SMEC TM packet	
Parameter	***Description***	***Value***
TelemetryPacket10SampFreq	*SMEC TM packet sampling frequency in units of scheduler cycles (1 cycle= 360 µs). 11 cycles (≡ 0x000b) correspond to a sampling frequency of 240 Hz*	*0x002c*

Command	**Description**	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run*	*0*

Command	**Description**	
DCU: SetSpectroMode	Set the LIA offsets for the Spectrometer channels automatically	
Parameter	***Description***	***Value***
Mode	*Indicates that the LIA offsets are selected automatically. Not clear what these mode definitions mean.*	*14*

Command	**Description**	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***

StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*1*

Command	**Description**	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*0*

Command	**Description**	
DCU: SetSpectroMode	Set the LIA offsets for the Spectrometer channels automatically	
Parameter	***Description***	***Value***
Mode	*Indicates that the LIA offsets are selected automatically. . Not clear what these mode definitions mean.*	*1c*

Command	**Description**	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms for the offsets to be sent to the DPU*	*1*

Command	**Description**	
DCU: SetStartFrame	Stop frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*0*

Command	**Description**	
DCU: SetDataMode	Set the DCU output data format between bolometer or 4-bit offset transmission and test pattern	
Parameter	***Description***	***Value***
DataMode	*00100 – Full Spectrometer (bit #2 is set to 1)*	*00100*

Command	**Description**	
DCU: SetFramenber or SetFrameCount?	Set frames acquisition mode	
Parameter	***Description***	***Value***
FrameCount or frame?	*0 – Continuous*	*0*

Command	**Description**	
DCU: SetStartFrame	Start frame generation	
Parameter	***Description***	***Value***
StartFrame	*0 – Stop; 1 – Run* *Wait xx ms*	*1*

- It is assumed here that the above sequence of acquisition and configuration commands can be executed before the DPU explicitly requests BDA data from the DCU. Until that happens there are no BDA data being generated.
- How is the home position found?
- This BB may need to be modified for the step-and-integrate Observatory Functions SOF3 and SOF4.

7.10 SCU: SCAL_OFF: Switch off SCAL

7.10.1 Interface

TBW

7.10.2 Implementation

There will be no specific command to switch the SCAL 2% and SCAL 4% on or off. Null SCAL current commands (≡ ±5 μA TBC) are equivalent to the off command.

7.11 PHOT_HDC_RESET: Reset Photometer Hardware and Data Configuration

Resets the hardware and data acquisition configuration at the end of an observation with the Photometer.

7.11.1 Interface

TBW

7.11.2 Implementation

TBW

7.12 SPEC_HDC_RESET: Reset Spectrometer Hardware and Data Configuration

Resets the hardware and data acquisition configuration at the end of an observation with the Spectrometer.

7.12.1 Interface

TBW

7.12.2 Implementation

TBW

7.13 MCU: BSM_OFF - Switch off the BSM

7.13.1 Interface

TBW

7.13.2 Implementation

Command	Description	
MCU: SetChopLoopMode	Open the chop loop and set the DAC to zero	
Parameter	***Description***	***Value***
ChopLoopMode	*0 - the chopper MRS position set to zero*	*0*

Command	Description	
MCU: SetCSensorPwr	Power off the MRS for the chopper axis	
Parameter	***Description***	***Value***

CSensorPwr	*0 – chopper sensor power off*	*0*

Command	**Description**	
MCU: SetJigLoopMode	Open the jiggle loop and set the DAC to zero	
Parameter	***Description***	***Value***
JigLoopMode	*0 - the jiggle MRS position set to zero*	*0*

Command	**Description**	
MCU: SetJSensorPwr	Power off the MRS for the jiggle axis	
Parameter	***Description***	***Value***
JSensorPwr	*0 – jiggle sensor power off*	*0*

7.14 Switch from any mode to SAFE mode

This operation is to be performed by the CDMS after direct request from the DPU.
TBW

7.15 Switch from SAFE to ON mode

TBW

8. BUILDING BLOCKS FOR THE SPIRE OBSERVING MODES

8.1 CHOP: Chop the BSM while taking BDA Data

8.1.1 Interface

Building Block	**Description**	
CHOP	Chop the BSM while taking BDA data	
Parameters	***Description***	***Value***
Y_0	*Initial BSM chop position (the "on source" position)*	
Y_1	*Final BSM chop position (the "off source " position)*	
N_{chops}	*Number of BSM chop cycles to be performed between y_0 and y_1*	
P_{chop}	*Period of BSM chop half-cycle, i.e. dwell time at y_0 (or y_1)*	
$N_{chopframes}$	*Number of BSM chop position frames per chop half-cycle*	
$N_{BDAframes}$	*Number of BDA frames per chop half-cycle*	
$P_{BDAframes}$	*Time period between the sampling of successive BDA frames*	
T_{BDA}	*Time delay between the issuing of the start chopping command and the sampling of BDA frames*	

8.1.2 Implementation

Command	**Description**

MCU: SetChopTargetPosition	Set up the chopper target position y	
Parameter	***Description***	***Value***
ChopTargetPosition	*Chopper MRS position (needs calibration table) ADC (16 bits)*	

Command	**Description**	
MCU: SetBSMMove	Move the BSM synchronously in both the chopper and jiggle axes	
Parameter	***Description***	***Value***
BSMMove	*Flag to indicate type of BSM movement* *1 – synchronous movement of both axes*	*1*

8.2 PCAL: Photometer Calibration with PCAL source

This BB includes all commands to the SCU for switching on the PCAL source, performing the Photometer calibration and then switching off the PCAL source. It is intended to monitor system gain or responsivity drifts.

8.2.1 Interface

Building Block	**Description**	
PCAL	Perform calibration measurement for the Photometer with PCAL.	
Parameters	***Description***	***Value***
N_{cycles}	*Number of calibration cycles to be executed*	
Mode	*Type of waveform for calibration cycle*	
Level	*Initial level of PCAL before start of calibration cycles*	
Amplitude	*Amplitude of PCAL waveform*	
N_{SCU}	*Number of SCU frames*	
P_{SCU}	*Time interval between SCU frames*	
N_{BDA}	*Number of BDA frames*	
P_{BDA}	*Time interval between BDA frames*	
T_{BDA}	*Time delay for collecting BDA frames*	

- It is expected that the detector data samples will be labelled with the sample number by the DCU.
- SCU parameters should also be labelled with the sample number within the frame.
- The OBS will have to keep track of the PCAL excitation number.

8.2.2 Implementation

There will be no specific command to switch the PCAL on or off. A null PCAL current command (≡ ±5 μA TBC) is equivalent to the off command.

8.3 PSCAN: Take Photometer data while the telescope is scanning

8.3.1 Interface

Building Block	**Description**	
PSCAN	Take Photometer BDA data while scanning the telescope	
Parameters	***Description***	***Value***
$N_{BDAframes}$	*Number of BDA frames*	

$P_{BDAframes}$	*Time period between the sampling of successive BDA frames*	
T_{BDA}	*Time delay between the issuing of PSCAN command and the sampling of BDA frames*	

8.3.2 Implementation

TBW

8.4 JIGGLE: Move the BSM to a given Jiggle Position

This is a very simple BB. Its function is to move the BSM to a jiggle position before other building blocks are executed (e.g. **CHOP**, **SPEC_SCAN**, etc). The jiggle position could be for any n-point jiggle table (n=7, 16, 25, 32 or 64).

8.4.1 Interface

Building Block	**Description**	
JIGGLE	Move the BSM to a given jiggle position	
Parameters	***Description***	***Value***
Y	*Y position of the BSM (Chop axis)*	*TBD*
Z	*Z position of the BSM (Jiggle axis)*	*TBD*
$T_{Duration}$	*Duration of the JIGGLE command - it is the time taken for the BSM to reach the given jiggle position and stabilise*	*TBD*

8.4.2 Implementation

Command	**Description**	
MCU: SetJigTargetPosition	Set up the jiggle target position z	
Parameter	***Description***	***Value***
JigTargetPosition	*Jiggle MRS position (needs calibration table) ADC (16 bits)*	

Command	**Description**	
MCU: SetBSMMove	Move the BSM synchronously in both the chopper and jiggle axes	
Parameter	***Description***	***Value***
BSMMove	*Flag to indicate type of BSM movement* *1 – synchronous movement of both axes*	*1*

8.5 SPEC_SCAN: Scan the FTS while taking detector data

8.5.1 Interface

Building Block	**Description**	
SPEC_SCAN	Scan the FTS while taking data	
Parameters	***Description***	***Value***
N	*Number of FTS scans to be performed*	
$N_{SMECframes}$	*Number of SMEC position frames per scan*	
$N_{BDAframes}$	*Number of BDA frames per scan*	

$P_{BDAframes}$	*Time period between the sampling of successive BDA frames*	
T_{BDA}	*Time delay between the issuing of the start scanning command and the sampling of BDAs*	

8.5.2 Implementation

TBW

8.6 SPEC_STEP: Step the FTS mirror

8.6.1 Interface

Building Block	**Description**	
SPEC_STEP	Step the FTS mirror to required position	
Parameters	***Description***	***Value***
N	*Number of steps the FTS mirror is to be moved*	*TBD*
Δx	*Step size in SMEC units*	*TBD*
T_{SMEC}	*Time delay between the issuing of the stepping command and its execution by the SMEC*	

8.6.2 Implementation

TBW

8.7 SPEC_PCAL: PCAL for FTS

8.7.1 Interface

This BB is similar to the one used for internally calibrating the Photometer. The intention is to also use the PCAL source for monitoring the FTS responsivity or gain drifts.
NOTE: At present there is no calibration Observatory Function for the FTS in [AD-1].

Building Block	**Description**	
SPEC_PCAL	Perform PCAL type calibration measurement for the FTS using the PCAL source	
Parameters	***Description***	***Value***
N_{cycles}	*Number of calibration cycles to be executed*	
Mode	*Type of waveform for calibration cycle*	
Level	*Initial level of PCAL before start of calibration cycles*	
Amplitude	*Amplitude of PCAL waveform*	
N_{SCU}	*Number of SCU frames*	
P_{SCU}	*Time interval between SCU frames*	
N_{BDA}	*Number of BDA frames*	
P_{BDA}	*Time interval between BDA frames*	
T_{BDA}	*Time delay for collecting BDA frames*	

8.7.2 Implementation

TBW