# DEC/MEC User Manual

# For OBS version 6.028

Doc. PACS-CL-SR-002, 4.8
11 November, 2008

Prepared by : B. Marquet, A. Mazy

Verified by :

Authorised by : E. Renotte

Approved by :

filename : PACS-CL-SR-002 SUM v4.8.doc

# Distribution List

| Recipients | Affiliation |
|---|---|
| CSL PACS Team c/o E. Renotte | CSL - Liege |
| PANDORA @ K.U. Leuven c/o W. De Meester | KUL - Leuven |
| Livelink Intranet @ astro.estec.esa.nl | ESTEC - Noordwijk |
| Herschel-Planck Project Team c/o A. Heske | ESTEC/SCI/PT - Noordwijk |
| A. Poglitsch, O. Bauer, R. Katterloher, N. Geis, B. Voss, H. Feuchtgruber, G. Wildgruber (Projectoffice MPE), Ch. Hartinger, W. Schmid | MPE - Garching |
| D. Lemke, R. Hofferbert, U. Grözinger, H. Dannerbauer, U. Klaas | MPIA - Heidelberg |

# Document Change Record

| Issue | Date | Comments |
|---|---|---|
| draft 0.1 | 09/08/01 | initial issue |
| draft 0.2 | 30/01/02 | updated HK list, included command list |
| draft 0.3 | 01/03/02 | updated the HK list :<br>• last BOLC entry is now 195 => every ID has been offset by 4.<br>• Added the Hk Entry 443 and 444.<br>• Split the DMC_LAST_ERR_BUF array into 16 entries |
| draft 0.4 | 4/03/02 | added 16 spare HK entries |
| draft 0.5 | 19/07/02 | Trigger commands changes :<br>• SwitchOnFilterWheelsControllers -> SwitchOnFWSpec<br>• SwitchOffFilterWheelsControllers->SwitchOnFWPhoto<br>Updated many command parameters description.<br>Bits affectation in DMC_XXX_STATUS has changed (where XXX is a task).<br>Added trigger command 85 (test command)<br>Additional error messages |
| issue 1.0 | 21/08/02 | New error code : 0x901<br>Changed the name of HK 447 & HK 448 (they had the same name as HK 445 & HK 446)<br>Added bit 20 in HK 201 & HK 204 |
| issue 2.0 | 11/09/02 | Added PID parameters description<br>Updated description of HK 208 to HK 214<br>Updated description of DEC_WRITE_TIMING_FPGA_PARAM<br>This issue of the User Manual concerns OBS version 2. |

| | | |
|---|---|---|
| Issue 2.1 | 13/12/02 | ECP-KUL-281002-02 : replaced the 'label' by the number of measures in the HK diagnostic packet (page45, byte22).<br>This issue of the User Manual concerns OBS version 4. |
| Issue 2.2 | 16/04/03 | Modified the definition of Write commands 150,151 and 152<br>Modified the definition of trigger command 59<br>Added error code 0x0210<br>Added the definition of END_OF_HK_LIST_ID<br>Renamed HK entry 223 (DMC_LABEL -> DMC_SEQ_LABEL) to avoid a name conflict between the Hk entry and the command.<br>Changed the layout of HK measure description (red text should be checked carefully in the MIB (bit position, name, …))<br>Changed HK and command names to match those in the MIB |
| Issue 2.3 | 10/06/03 | Modified the definition of Hk Entry n°450.<br>Updated description of Hk Entries 226 and 227.<br>Renamed all the spare fields in status HK (e-mail from Milena on 9/5)<br>Added trigger commands n°86 -> 89. |
| Issue 2.4 | 30/07/03 | Modified the name of trigger commands 17 and 18.<br>Update description of write commands 158 and 159<br>Added more information on failure codes returned by trigger commands<br>Added more information on how to use the commands |
| Issue 2.5 | 03/10/03 | Modified the DMC_LOCK_GRAT command (now has a parameter).<br>Added a new valid parameter value for DMC_START_DIAG_HK<br>Modified the HK Entries 452, 453 and 454.<br>Modified the description of HK Entries 238, 239, 240, 463.<br>Modified the naming of counters in Diagnostic Hk packet header<br>Added section 5.1 about time-stamping.<br>All these changes will be reflected in EM software only. MIB can be updated immediately and will remain compatible with AVM software.<br>Modified DMC_WRT_SPU_TRAN_MODE which now takes 2 parameters.<br>Modified the HK Entries 450 and 451<br>These changes will be reflected in AVM software. |
| Issue 2.6 | 05/03/04 | Modified HK Entries 455 -> 458, 208, 210, 211, 255 -> 264<br>Renamed HK entry 223, 237<br>Added HK Entries with ID >= 512<br>Modified Trigger commands 42, 44, 45, 46, 47, 48<br>Removed Trigger commands 60, 61, 62, 63<br>Added section 'Detecting Memory Errors'<br>Removed Activity ID 149: DMC_WRT_COOLER_CONF_PAR<br>Added the chopper position offset in DMC_WRT_CHOP_CONF_PAR |

| | | |
|---|---|---|
| | | Added Activity ID 160: DMC_WRT_GRAT_REDUNDANT<br>Added Activity ID 161: DMC_WRT_GRAT_RANGE<br>Modified Activity ID 145 and 146<br>Added section "Dump Addresses"<br>Added section "working with the redundant grating"<br>Modified the format of the Hk Diag Packet |
| Issue 2.7 | 19/05/04 | Added trigger commands 90-96,<br>Renamed trigger commands 71 and 75.<br>Added conversion information for all DEC HK + 405->408, 427, 428, 429<br>Modified HK entries 208, 209, 213, 214, 243->246, 259, 267, 278, 283, 295, 296, 301, 312, 335, 346, 369, 380, 404, 445->448, 453, 459, 460, 512->521<br>Modified Activity ID 147, 148, 157<br>Removed error codes 0xB12, B22, B32, B42, B52, B62<br>Note : this SUM is a preliminary version for the EM. A few changes might still occur on EM OBS. The AVM OBS will never be adapted to implement what is documented here. |
| Issue 2.8 | 26/05/04 | Modified HK entries 208 (bit24), 265, 270, 272, 299, 304, 306, 333, 338, 340, 367, 372, 374<br>Changed conversion units for HK entries :265->280, 299->314, 333->348, 367->382<br>Modified Activity ID 157.<br>Note : this SUM is a preliminary version for the EM. A few changes might still occur on EM OBS. The AVM OBS will never be adapted to implement what is documented here. |
| Issue 3.0 | 07/07/04 | Modified HK entries 413, 414, 571, 572, 575, 576, 291, 325, 359, 393<br>Modified conversion units for HK entries 292, 293, 326, 327, 360, 361, 394, 395<br>Modified parameters value for trigger ID 57<br>Added a description for bit9 of CRE_Ctrl_Reg<br>Added a section "Connection/reconnection strategy"<br>Added some information about the duration of trigger commands<br>Updated section about time-stamping to take into account the replacement of DMC_5OBT_COUNT by DMC_OBT_COUNT<br>Documented the default values for DEC FPGA |
| Issue 3.1 | | Modified Activity ID 143, 144, 154, 155<br>Modified Trigger ID 57<br>Added a section about temperature sensors<br>Updated the section about the grating<br>Changed the units for HK entries 295, 296, 329,330, 363, 364, 397, 398, 405->408, 426, 427, 429 |
| Issue 3.2 | 15/10/04 | Changed length of Activity ID 147, 148 |

| | | |
|---|---|---|
| | | Changed HK entries 419, 522->553, renamed bit10-15 in hk359,<br>Changed units in Hk entries286, 354, 320, 321, 388, 389, 409->412, 420, 424, 425, 428, 430, 554->578<br>Changed the units for Trigger ID 17, 18<br>Removed useless parameters for Trigger ID 47, 48<br>Renamed Hk Entry 364 + all renaming suggested by Milena<br>Non destructive sync is always 2 CRE clock width =><br>Changed bit8 in Hk entries 291, 325, 359, 393 and changed the definition of CRE_Ctrl reg in Activity ID 154 and 155 |
| Issue 3.3 | 06/12/04 | Modified section about grating<br>Added procedure to upload new version of the OBS<br>Modified Hk entry 512, 513, 514, 515<br>Modified default parameters for chopper and grating.<br>Additional information provided concerning activity ID 162 |
| Issue 3.4 | 28/04/05 | Renamed bit2 and bit3 in CRE_CTRL_REG<br>Modified bit2 and bit3 meaning in Hk entry 291, 325, 359, 393<br>Renamed bit3 and 5 of Timing FPGA control register<br>Added section about Synchronization of DMC science header and science data<br>Added error code 0x0B24<br>Additional information provided concerning HK coming from SPU (HK 419 -> 427) |
| Issue 4.0 | 07/02/06 | This version of the SUM relates to software that will be installed on QM and FM DMC only.<br>Added a section on 'how to determine the grating range'<br>Modified section about 'Using the spectroscopy detectors'<br>Added bit10&11 definition in Cre_ctrl_reg<br>Added bit 18 description in HK 196<br>added/updated commands for heater and flasher<br>updated HK related to heater and flasher<br>Changed the format of trigger parameters to UINT32 or INT32 (no more 8bits or 16bits parameters).<br>The CRE output conversion has been slightly modified<br>+ everything in red and in green<br>Introduced HK limit checking for HK ID 263, 264, 284, 352, 413, 414 |
| Issue 4.1 | | Unofficial release |
| Issue 4.2 | 09/02/07 | Changes from 4.0:<br>Everything that is marked in red.<br>Removed simplified PID chopper controller (bit2 of DMC_SELECT_MECH_CTRL_MODE) |
| Issue 4.3 | 05/06/07 | Updated FW commanding to take into account the 4 threshold values and the direction of rotation. Updated: HK Ids 210&211, Trigger ID64&66, Write commands 145&146 |

| | | |
|---|---|---|
| | | Updated Chopper commanding to consider the field plate lookup tables.<br>Other changes about chopper output filter description.<br>Updated the cre output conversion (6.27V instead of 6.22V)<br>Added section about Internal sampling frequency of housekeeping values<br>All changes are in red |
| Issue 4.4 | 07/06/08 | Added a new write command for the new filter in the grating controller<br>Updated the section about dump addresses (that can also be used to program the custom hk entries). |
| Issue 4.5 | 30/07/08 | Added new values for SelectFieltPlateLUT in the chopper parameters block.<br>Updated the section about dump addresses (that can also be used to program the custom hk entries).<br>Swapped definition of bit 10 & 11 of HK entries 291, 325, 359, 939 |
| Issue 4.6 | 07/10/08 | Added new parameters value for DMC_SYNCHRONIZE_ON_DET<br>Added new write-command DMC_WRT_GRAT_MAX_POWER<br>HK entry 234 is now a synchro counter |
| Issue 4.7 | 09/10/08 | Changed the id of the new write-command DMC_WRT_GRAT_MAX_POWER<br>DMC_SYNC_COUNT moved from hk entry 234 to 240.<br>DMC_CUSTOM_HK1 is now referencing the CRDCCP by default<br>New parameters values for the DMC_SELECT_MECH_MODE |
| Issue 4.8 | 10/11/08 | Corrected and updated definition of parameters values for the DMC_SELECT_MECH_MODE<br>Updated the definition of phase_shift_reg<br>Added section 4.4.21 describing the diagnostic mode |
| | | |

*last saved by Alain Mazy on 11-Nov-08*

# List of Abbreviations

| | |
|---|---|
| AC | Alternating Current |
| ADC | Analog-to-Digital Converter |
| AIV | Assembly Integration & Verification |
| AVM | Avionic Verification Model |
| ASW | Application SoftWare |
| BOLA | Bolometer Amplifier |
| BOLC | Bolometer Controller |
| CoI | Co-investigator |
| CQM | Cryogenic Qualification Model |
| CSL | Centre Spatial de Liège |
| DAC | Digital-to-Analog Converter |
| DEC/MEC | Detector & Mechanism Controller |
| DC | Direct Current |
| DM | Data Memory |
| DMC | DEC/MEC |
| DPU | Digital Processing Unit |
| EEPROM | Electrically Erasable PROM |
| EGSE | Electrical Ground Support Equipment |
| EM | Engineering/Electrical Model |
| EMC | Electro-Magnetic Compatibility |
| FM | Flight Model |
| FPGA | Field Programmable Gate Array |
| FPU | Focal Plane Unit |
| FS | Flight Spare |
| HK | HouseKeeping |
| HW | Hardware |
| ICD | Interface Control Document |
| IID-A | Instrument Interface Document - Part A |
| IID-B | Instrument Interface Document - Part B |
| ISR | Interrupt Service Routine |
| NA | Not Applicable |
| OBS | On-Board Software |
| OBSW | On-Board SoftWare (=OBS) |
| PACS | Photodetector Array Camera and Spectrometer |
| PI | Prime Investigator |
| PM | Program Memory |
| PROM | Programmable ROM |
| QM | Qualification Model |
| RAM | Random Access Memory |
| ROM | Read-Only Memory |
| S/C | SpaceCraft |
| SPU | Signal Processing Unit |
| S/S | Sub-System |
| SSD | Software Specification Document |
| SUSW | StartUp SoftWare |
| TBC | To Be Confirmed |
| TBD | To Be Defined |
| URD | User Requirement Document |

# Table of Contents

# 1 Scope

## 1.1 Introduction

The Photodetector Array Camera and Spectrometer (PACS) is an imaging spectrometer-photometer which forms part of the science payload of the Herschel Space Observatory (formerly called FIRST), an ESA cornerstone mission (CS4) to be launched in 2007 on Ariane 5.

A presentation of the Herschel mission and status is available at URL: http://sci.esa.int/home/herschel/. Useful information on PACS instrument, mission and Consortium can be found at http://pacs.mpe-garching.mpg.de and http://pacs.ster.kuleuven.ac.be/.

## 1.2 Purpose

This document is the User Manual of DMC. It is targeted to ground operators, DPU software designer and DMC testers.

This version of the SUM relates to software that will be installed on QM and FM DMC only.

## 1.3 Organisational Responsibilities

The PACS project activities including project management and system engineering will be done at MPE-Garching under the direction of A. Poglitsch (PI). Design, fabrication, testing, and integration of the flight units will be done at CoI and commercial facilities as appropriate.

In this programme the CSL is responsible for the design, production and unit-level verification of:

- the focal plane Grating Assembly;
- the Detector & Mechanism Controller (DEC/MEC);
- the Warm Interconnecting Harness.

# 2 Documents

## 2.1 Applicable Documents

| | | |
|---|---|---|
| [AD1] | ESA PT-IID-A-04624 | FIRST/PLANCK Instrument Interface Document - Part A |
| [AD2] | ESA PT-RQ-04410 | PA Requirements for FIRST/PLANCK Scientific Instruments |
| [AD3] | PACS-ME-RS-004 | PACS Science Requirements Document |
| [AD4] | PACS-ME-RS-005 | PACS Instrument Requirements Document |
| [AD5] | PACS-ME-PL-007 | PACS Project Product Assurance Plan |
| [AD6] | PACS-CL-ID-003 | ICD DEC/MEC-DPU issue 3.5 from 3 october 2003 |

| | | |
|---|---|---|
| [AD7] | PACS-CL-ID-004 | ICD DEC/MEC-SPU issue 3.5 from 29 july 2004 |

## 2.2 Reference Documents

| | | |
|---|---|---|
| [RD1] | ESA PT-PACS-02126 | Instrument Interface Document - Part B - Instrument "PACS" |
| [RD2] | PACS-ME-PL-002 | PACS Design, Development and Verification Plan |
| [RD3] | HPL-MA-1248-02-CRS | CRISA DSP Module QM User's manual issue 1 from 22 sept 2005 |

# 3 Starting the software

## 3.1 Starting the software from EEPROM

### 3.1.1 SUSW self test

After a power-on of the DMC, the LLSW starts, performs a self test and then waits for commands. Allow 15 seconds for the LLSW to finish its start-up procedure.

You should then dump the result of the self test and check that it is correct. The address, size and expected results are (a detailed description of the expected result can be found in [RD3]):

| Address (DM) | Content | Expected result |
|---|---|---|
| 00000000 | Reset source | 00000001 |
| 00000001 | Hardware initialization tests result | 0007FFFF |
| 00000002 | PROM initialization checksum – computed value | 55D0 |
| 00000003 | PROM initialization checksum – expected value | 55D0 for SUSW 1.1 |
| 00000004 | Error detection during command loop execution | 00000004 |
| 00000005 | EDAC double failure last wrong address – PM | 00000000 |
| 00000006 | EDAC double failure last wrong address – DM | 00000000 |
| 00000007 | DSP interrupt pending register value | 00000020 |
| 00000008 | PMPSC interrupt pending register value | 00000000 |
| 00000009 | DMPSC interrupt pending register value | 00000000 |
| 0000000A | SMCS interrupt pending register value | 00000000 |

#### 3.1.1.1 Note on PROM failures

During the module tests at Crisa, a problem in the contents of the PROM devices was detected. The problem has shown stable from that moment and does not represent a problem for the operation of the DMC SUSW. The only current effect (stabilized) is that the obtained checksum, as part of the boot selftest, is not as expected (55D0) but is 85C6, F557, F0C3, 5D44 or 2841.

A further failure in the same way cannot be discarded. It is possible that, if the failure appear, it has no effect in the operation of the software as the current failure. In this case, thee boot software obtained checksum would be different in the report area.

So, the recommended steps to be aware of the problem are:

- verify the boot selftests word 2, PROM Checksum computed value, in each switch-on. In case the checksum is not one of the expected one, dump the complete contents of the PROM by means of the boot software command loop in order to analyse the failure. Note that a modification of a single bit produces an absolutely different checksum word.
- Depending on the failed bit, it would event be not possible to dump the PROM contents.
- If the PROM contents can be obtained, Crisa will analyse the failed bit and the potential effects.
- If the failure does not affect operation, the unit will remain as it was.
- If the failure could affect operation, the failed memory will be replaced.

### 3.1.2 Copy OBS from EEPROM

Then, the OBS must be copied from EEPROM to PRAM. There are 2 segments to copy so 2 commands to send (allow a 500msec interval between the 2 commands) :

- Copy SEG_INIT:
  - Activity ID : 0x65
  - SID : 5
  - Parameter 1 (mem ID of EEPROM) : 3
  - Parameter 2 (start address in EEPROM) : 0
  - Parameter 3 (mem ID of PRAM) : 1
  - Parameter 4 (start address in PRAM): 0x6EE00
  - Parameter 5 (length): 0x4000
- Copy SEG_PMCO:
  - Activity ID : 0x65
  - SID : 5
  - Parameter 1 (mem ID of EEPROM) : 3
  - Parameter 2 (start address in EEPROM) : 0x8000
  - Parameter 3 (mem ID of PRAM) : 1
  - Parameter 4 (start address in PRAM): 0x8000
  - Parameter 5 (length): 0x8000

### 3.1.3 Give control to ASW

Then, the next step is to give the control to the application software:

- Give Control to APSW:
  - Activity ID : 0x66
  - SID : 2
  - Parameter 1 (mem ID of PRAM) : 1
  - Parameter 2 (start address in PRAM) : 0x8032

Then, allow 6 seconds to let the OBS start, send the first hk packets and wait for the first command.

## 3.2 Building a new version of the software

Note that this procedure should be used by CSL only. You should not try to rebuild a new version of the software by yourself. It is provided here as a reminder for CSL.

1. In params.c, change the version number
2. Rebuild the executable (delete all object files and build)
3. execute “split.bat” to generate segment files
4. execute “upload_obs.exe” to generate the uploadable file
    a. build files for seg_init
    b. build files for seg_pmco
    c. convert files from binary to ascii
    d. generate the tcl script file to check the memory load.

## 3.3 Uploading a new version of the software

Each time a new version of the software is available, CSL will provide:

- A set of TC to upload the new software in RAM
- A tcl script to check that the memory load has succeeded

The telecommands must be sent to DMC SUSW. The procedure to upload a new version is:

1. switch-on DMC
2. Let it run for 15 seconds
3. send the TC to upload the new version
4. execute the tcl script to check the memory if all tests are successful, continue the procedure
5. Give control to APSW (same command as in previous section). Do not copy EEPROM into RAM !

6. Let the HLSW start as usual
7. Send the DMC_COPY_OBS_TO_EEPROM command to copy the software from RAM to EEPROM. This should be done only when the CPU load is low and when no mechanisms are controlled.

### 3.3.1 Summary of HLSW commands and telemetry

**Trigger Commands:**

- 88 DMC_COPY_OBS_TO_EEPROM

**Write Commands:**

- none

**HK nominal:**

- 243 DMC_VID
- 196 DMC_SW_GLOBAL_ST

**HK diag:**

- none

## 3.4 Interface with other warm electronics subsystems

### 3.4.1 1355 communication handling by the DMC OBS

The DMC OBS handles 6 1355 links. For each of this link, there is a task that is dedicated to each of the direction. The task that handle reception is generally called 'receiver' and the task that handles the emission is called 'controller', 'sender' or 'encoder'.

For the interface with DPU, there is a 'DPU Receiver' task that receives all telecommands from DPU and another task 'DPU Sender' that is sending the acknowledges to these commands and that is sending the housekeeping packets.

For the interface with DEC, there is a 'DEC Controller' task that configures the DEC by sending commands and there is a 'DEC Receiver' task that receives all the science data packets.

For the interface with BOLC, there is a 'BOLC Controller' task that sends the telecommands to BOLC and there is a 'BOLC Receiver' task that receives all the science data packets.

For the interface with SPU, since the communication is uni-directionnal, there is only a 'Packet Encoder' task for each SPU.

Each of these tasks has a status word in the nominal housekeeping that gives information about the status of the 1355 connection (connected or not). Each of these tasks also has a packet counter to monitor the traffic on the link.

### 3.4.2 Summary of HLSW commands and telemetry

**Trigger Commands:**

- 86 DMC_START_RED_SPU_LINK
- 87 DMC_START_BLUE_SPU_LINK
- 89 DMC_RESET_SMCS_CHIP_2

**Write Commands:**

- 150 DMC_WRT_BOL_REC_OPT
- 151 DMC_WRT_B_DEC_REC_OPT
- 152 DMC_WRT_R_DEC_REC_OPT
- 158 DMC_WRT_B_PACKT_ENC_LINK
- 159 DMC_WRT_R_PACKT_ENC_LINK

**HK nominal:**

- 198 DMC_DPU_REC_STAT
- 199 DMC_DPU_SEN_STAT
- 200 DMC_DECB_REC_STA
- 201 DMC_DECB_CTRL_ST
- 202 DMC_BLUE_PAC_ENC
- 203 DMC_DECR_REC_STA
- 204 DMC_DECR_CTRL_ST
- 205 DMC_RED_PAC_ENC
- 206 DMC_BOL_REC_STAT
- 207 DMC_BOL_CTRL_STA
- 228 DMC_DECB_REC_PAC
- 229 DMC_DECR_REC_PAC
- 230 DMC_DECB_CTRL_PA
- 231 DMC_DECR_CTRL_PA
- 232 DMC_BLUE_ENC_PAC
- 233 DMC_RED_ENC_PAC
- 234 DMC_BOL_REC_PAC
- 235 DMC_BOL_CTRL_PAC
- 236 DMC_DPU_REC_PAC
- 237 DMC_DPU_SEND_PAC

**HK diag:**

- none

### 3.4.3 Interface with DPU

At startup of the ASW, a reset of the SMCS chip 1 (the one connected to DPU and both SPUs) is performed. After that, the ASW does not do anything for 6 seconds and then, initiate the link with DPU as master. If this connection fails, there is no retry.

Afterwards, as soon as a disconnection with DPU is observed, the ASW enters a reconnection loop. Every 9 seconds, the chip is reset and the communication with DPU is started again (as master). The first chip reset occurs only 9 seconds after the disconnection has been detected.

### 3.4.4 Interface with SPUs

The communication with SPU is initiated only on request (by trigger command). Two commands are available to start the 1355 communication with the SPUs. The master/slave status can be chosen for this commands but it is recommended to use DMC as master in order to complete the nominal PACS switch on procedure.

### 3.4.5 Interface with BOLC

After a BOLC switch-on, a chip reset of the SMCS 2 should be performed. This will also start the communication with BOLC (DMC as master). If DECs are already powered on and connected, the chip reset will interrupt the communication. Therefore, a few packets from DEC will be lost. At that time, the DEC receiver and controller will observe the disconnection and will raise an error in their status words.

If an unexpected disconnection is observed on BOLC link, the DMC OBS will simply signal the error in the BOLC receiver and controller tasks. It will not try to reconnect by itself. DPU should send the DMC_RESET_SMCS_CHIP_2 to try to resume the connection.

### 3.4.6 Interface with DECs

When you send a command to power-on a DEC, the power is supplied immediately. The DEC then needs 5 seconds to initialize. The DMC HLSW then resets the SMCS 2 and connects to the DEC that has been powered on and to the BOLC and the other DEC if they were already connected before. At that time, the other DEC receiver and controller and BOLC receiver and controller will observe the disconnection and will raise an error in their status words.

If an unexpected disconnection is observed on DEC link, the DMC OBS will simply signal the error in the DEC receiver and controller tasks. It will not try to reconnect by itself. DPU should send the DMC_RESET_SMCS_CHIP_2 to try to resume the connection.

### 3.4.7 Master/Slave strategy

For all links, DMC should be the Master. Although it can be configured as slave for the communication with SPUs.

# 4 Commands

## 4.1 Trigger Commands

Below, you will find the list of trigger commands. The format trigger command is described in [AD6]. The commands are categorized as follows by their validity :

| | |
|---|---|
| S | Command accepted only in sequence files |
| T | Command accepted only as trigger |
| A | Command accepted in sequence and as trigger |

| Validity | Symbol | Activity ID | Parameter Type | Parameter Description | Subsystem |
|---|---|---|---|---|---|
| S | DMC_LOOP | 0 | UINT32 | number of repetitions | Sequencer |
| S | DMC_END_LOOP | 1 | NONE | Last instruction of a loop. | Sequencer |
| S | DMC_WAIT | 2 | UINT32 | number of time units to wait for. In spectroscopy mode, time unit is a ramp; in photometry mode, time unit is a readout | Sequencer |
| S | DMC_END_SEQUENCE | 3 | NONE | Last instruction of a sequence. | Sequencer |
| S | DMC_LABEL | 4 | UINT32 | Label ID. 8 bits used to identify the position in the sequence. This, combined with the sequence ID (see below) and the readout id, is included in the data packets sent to the SPU to identify the readout. | Sequencer |
| T | DMC_START_SEQUENCE | 5 | NONE | Start the execution of the sequence previously uploaded by the DEC_WRITE_SEQUENCE command. | Sequencer |
| T | DMC_ABORT_SEQUENCE | 6 | NONE | Abort the execution of the sequence currently being executed. After an abort, the sequence is ready to start again (at its beginning) | Sequencer |
| T | DMC_SET_TIME | 7 | NONE | Set the time previously written by the DEC_WRITE_TIME command. | Time Stamping |
| T | DMC_SET_OBSID | 8 | UINT32 | Sets the Observation ID | Time Stamping |
| T | DMC_SET_BBID | 9 | UINT32 | Sets the Building Block ID | Time Stamping |
| T | DMC_SYNCHRONIZE_ON_DET | 10 | UINT32 | Select the synchronization source for the Sequencer. Note that this synchro source is also used to trigger the mechanism movement and that the DMC_OBT_COUNT is updated only when this synchro signal is received. PARAM : | Synchro |

| | | | | | |
|---|---|---|---|---|---|
| | | | | In the 8 lsb, set the synchronization source for the Sequencer :<br>1 = Synchronize on blue spectro ramps (and mechanisms use the sync to start their move)<br>2 = Synchronize on red spectro ramps (and mechanisms use the sync to start their move)<br>4 = Synchronize on BOL readouts (and mechanisms use the sync to start their move)<br>9 = Synchronize on blue spectro ramps (and mechanisms don't use the sync to start their move)<br>10 = Synchronize on red spectro ramps (and mechanisms don't use the sync to start their move)<br>12 = Synchronize on BOL readouts (and mechanisms don't use the sync to start their move)<br>If bit 11 is set to 0, the timing FPGA will use the default synchronization sources as defined above,<br>If bit 11 is set to 1, the timing FPGA will use one of the following synchronization sources :<br>In bits 8-10, set the synchronization source for the timing FPGA :<br>000 = internal programmable generator (nominal = 256Hz)<br>001 = Red DEC supply group 1<br>010 = Red DEC supply group 2<br>011 = Blue DEC supply group 3<br>100 = Blue DEC supply group 4<br>101 = BOLC<br>110 = internal programmable generator (nominal = 40Hz)<br>111 = spare (external generator only for ground testing)<br>Note : by default, sequencer synchronizes on the<br>Blue Spectrometer. | |
| T | DMC_SET_TIMING_FPGA_PAR | 11 | NONE | Copy the parameters previously written by the DEC_WRT_TIMING_FPGA_PAR command in the FPGA registers | Synchro |
| T | DMC_SWON_B_DEC | 12 | NONE | Switch on Blue DEC electronic power-supply, wait for 1355 link to initialize (DEC sends packet : valid HK + invalid data (detectors are off) )<br>DURATION : 8 seconds<br>FAILURE CODE :<br>0xAA : the other CREs are switched-on | DEC blue |
| T | DMC_SWOF_B_DEC | 13 | NONE | Switch off Blue DEC electronic power-supply | DEC blue |
| T | DMC_SWON_B_SPEC | 14 | NONE | Switch on detector array power-supply (data are read on the detector)<br>DURATION : 15 seconds<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SWOF_B_SPEC | 15 | NONE | Switch off detector array power-supply<br>DURATION : 15 seconds<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SET_PAR_B_SPEC | 16 | NONE | Send complete parameters table to blue DEC | DEC blue |

| | | | | | |
|---|---|---|---|---|---|
| | | | | FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | |
| T | DMC_SET_B_SPEC_HEAT_C | 17 | UINT32 | 12bits used to represent the Blue DEC heater current (0=0mA, 4095=20mA)<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SET_B_SPEC_FLASH_C | 18 | UINT32 | 12bits used to represent the Blue DEC flasher current (0=0mA, 4095=20mA)<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SWON_R_DEC | 19 | NONE | Switch on Red DEC electronic power-supply, wait for 1355 link to initialize (DEC sends packet : valid HK + invalid data (detectors are off) )<br>DURATION : 8 seconds<br>FAILURE CODE :<br>0xAA : the other CREs are switched-on | DEC red |
| T | DMC_SWOF_R_DEC | 20 | NONE | Switch off Red DEC electronic power-supply | DEC red |
| T | DMC_SWON_R_SPEC | 21 | NONE | Switch on detector array power-supply (data are read on the detector)<br>DURATION : 15 seconds<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| T | DMC_SWOF_R_SPEC | 22 | NONE | Switch off detector array power-supply<br>DURATION : 15 seconds<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| T | DMC_SET_PAR_R_SPEC | 23 | NONE | send complete parameters table to red DEC.<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| T | DMC_SET_PAR_BOTH_SPEC | 24 | NONE | send parameters tables to both DECS with master reset to ensure synchronisation (restrictions apply on the respective parameters values)<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DECs |
| A | DMC_VAL_SCI_DATA_B | 25 | NONE | Validate the Blue science data (from DEC and/or BOLC) | Science Data |
| A | DMC_VAL_SCI_DATA_R | 26 | NONE | Validate the Red science data (from DEC and/or BOLC) | Science Data |
| A | DMC_VAL_SCI_DATA_BOTH | 27 | NONE | Validate the Blue and Red science data (from DEC and/or BOLC) | Science Data |
| A | DMC_INVAL_SCI_DATA_B | 28 | NONE | Invalidate the Blue science data (from DEC and/or BOLC) | Science Data |
| A | DMC_INVAL_SCI_DATA_R | 29 | NONE | Invalidate the Red science data (from DEC and/or BOLC) | Science Data |
| A | DMC_INVAL_SCI_DATA_BOTH | 30 | NONE | Invalidate the Blue and Red science data (from DEC and/or BOLC) | Science Data |

| | | | | | |
|---|---|---|---|---|---|
| T | DMC_START_DET_SIMULATO R | 31 | UINT32 | 4 MSb : detector id (1 = BLUE_SPEC, 2 = RED_SPEC, 4 = BOLC (can be a combination of bits)), 28 LSb : period.<br>See section "Using the Commands : Simulating detectors" | Detectors |
| T | DMC_STOP_DET_SIMULATOR | 32 | NONE | Stops the detector simulator | Detectors |
| T | DMC_SEND_COMMAND_TO_ BOLC | 33 | UINT32 | Send a command to BOLC. The parameter is the command.<br>FAILURE CODE :<br>0xAA : The connection between DMC and BOLC is not established | BOLC |
| T | DMC_SET_R_SPEC_HEAT_C | 34 | UINT32 | 12bits used to represent the Red DEC heater current (0=0mA, 4095=20mA)<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| T | DMC_SET_R_SPEC_FLASH_C | 35 | UINT32 | 12bits used to represent the Red DEC flasher current (0=0mA, 4095=20mA)<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| T | DMC_SPARE_CMD_1 | 36 | UINT32 | Spare Command<br>FAILURE CODE :<br>0xA7 : Invalid trigger ID | |
| T | DMC_RESET_BOL_READOUT _C | 37 | NONE | Resets BOLC Readout Counter | Time Stamping |
| T | DMC_SWON_GRAT_CONT | 38 | NONE | Switch on power supply (Drive + Inductosyn) | Grating |
| T | DMC_SWOF_GRAT_CONT | 39 | NONE | Switch off power supply (Drive + Inductosyn) | Grating |
| T | DMC_ENABLE_GRAT_CONT | 40 | NONE | Activate servo-loop, copy current position in target<br>FAILURE CODE :<br>0xAA : Grating is not switched-on | Grating |
| T | DMC_DISABLE_GRAT_CONT | 41 | NONE | Deactivate servo-loop, output = 0 (no torque) | Grating |
| A | DMC_MOVE_GRAT_ABS | 42 | INT32 | Move grating to absolute position.<br>PARAM : Target position<br>If in closed loop mode :<br>Target position is in DMC_GRAT_CUR_POS units<br>If in open loop mode :<br>Target position is in DMC_GR_DEG_POS units<br>DURATION : Setpoint is incremented until target is reached depending on grating controller RATE parameter value<br>FAILURE CODE :<br>0xAA : Grating controller is not enabled or homing has not been done | Grating |
| A | DMC_MOVE_GRAT_REL | 43 | INT32 | Move grating to relative position.<br>PARAM : Target position = relative move from current position. End position = current position + target position<br>If in closed loop mode :<br>Target position is in DMC_GRAT_CUR_POS units<br>If in open loop mode :<br>Target position is in DMC_GR_DEG_POS units<br>DURATION : Setpoint is incremented until target is reached depending on grating controller RATE parameter value | Grating |

| | | | | | |
|---|---|---|---|---|---|
| | | | | FAILURE CODE :<br>0xAA : Grating controller is not enabled | |
| T | DMC_HOME_GRAT | 44 | UINT32 | Search for the hard stop and initialize the inductosyn period counter correctly.<br>PARAM :<br>0 : Move towards negative positions<br>1 : Move towards positive positions<br>DURATION : max 70 seconds (if the Grating PID rate is 3) depending on the initial position of the grating.<br>FAILURE CODE :<br>0xAA : Grating controller is not enabled | Grating |
| T | DMC_ENTER_GRAT_CONT_D EG | 45 | UINT32 | Enter grating degraded mode.<br>PARAM :<br>0 : Open-loop mode<br>Other modes might come later<br>NOTE :<br>This command must be sent after the controller is switched on<br>FAILURE CODE :<br>0xAA : Grating controller is enabled | Grating |
| T | DMC_EXIT_GRAT_CONT_DEG | 46 | NONE | Exit grating degraded mode.<br>NOTE :<br>This command must be sent when the controller is switched off | Grating |
| T | DMC_LOCK_GRAT | 47 | UINT32 | Locks the grating.<br>PARAM :<br>0x2 : activate mechanical launch-lock motor 1 for 40960 ISR period<br>0x10 : activate mechanical launch-lock motor 2 for 40960 ISR period<br>0x12 : activate mechanical launch-lock motor 1+2 for 40960 ISR period<br>DURATION : 40960 ISR period (nominally 5 seconds)<br>NOTE :<br>Motor 1 = connected to currently active electronics<br>Motor 2 = connected to currently inactive electronics<br>Nominal operation uses motor 1+2<br>FAILURE CODE :<br>0xAA : Grating is not switched-on | Grating |
| T | DMC_UNLOCK_GRAT | 48 | UINT32 | Unlocks the grating.<br>PARAM :<br>0x8 : activate mechanical launch-lock motor 1 for 40960 ISR period<br>0x20 : activate mechanical launch-lock motor 2 for 40960 ISR period<br>0x28 : activate mechanical launch-lock motor 1+2 for 40960 ISR period<br>DURATION : 40960 ISR period (nominally 5 seconds)<br>NOTE :<br>Motor 1 = connected to currently active electronics<br>Motor 2 = connected to currently inactive electronics<br>Nominal operation uses motor 1+2<br>DURATION : 40960 ISR period (nominally 5 seconds)<br>FAILURE CODE :<br>0xAA : Grating is not switched-on | Grating |

| | | | | | |
|---|---|---|---|---|---|
| T | DMC_SWON_CHOP_CONT | 49 | NONE | Connect chopper to amplifier (release launch lock)<br>Set coils selection relays to nominal = 3 coils used | Chopper |
| T | DMC_SWOF_CHOP_CONT | 50 | NONE | disconnect chopper from amplifier (connect damping resistor)<br>Set coils selection relays to power off state | Chopper |
| T | DMC_ENABLE_CHOP_CONT | 51 | NONE | Activate servo-loop, copy current position in target<br>FAILURE CODE :<br>0xAA : Chopper is not switched-on | Chopper |
| T | DMC_DISABLE_CHOP_CONT | 52 | NONE | Deactivate servo-loop, output = 0 (no torque) | Chopper |
| A | DMC_MOVE_CHOP_ABS | 53 | INT32 | Move chopper to absolute position.<br>PARAM :<br>In closed loop :<br>Target position is in DMC_CHOP_CUR_POS units, i.e.<br>-32767 = -10V, 32767 = 10V<br>In open loop :<br>Target is the commanded current in the DAC and is therefore in DMC_CHOP_OUTPUT units, i.e.<br>-32767 = -133 mA, 32767 = 133 mA<br>FAILURE CODE :<br>0xAA : Chopper controller is not enabled | Chopper |
| A | DMC_MOVE_CHOP_REL | 54 | INT32 | Move chopper relative.<br>PARAM :<br>In closed loop :<br>Target position is in DMC_CHOP_CUR_POS units, i.e.<br>-32767 = -10V, 32767 = 10V<br>In open loop :<br>Target is the commanded current in the DAC and is therefore in DMC_CHOP_OUTPUT units, i.e.<br>-32767 = -133 mA, 32767 = 133 mA<br>FAILURE CODE :<br>0xAA : Chopper controller is not enabled | Chopper |
| A | DMC_MOVE_CHOP_ABS_DITHER | 55 | INT32 | Move chopper to absolute position + dither.<br>PARAM :<br>In closed loop :<br>Target position is in DMC_CHOP_CUR_POS units, i.e.<br>-32767 = -10V, 32767 = 10V<br>In open loop :<br>Target is the commanded current in the DAC and is therefore in DMC_CHOP_OUTPUT units, i.e.<br>-32767 = -133 mA, 32767 = 133 mA<br>FAILURE CODE :<br>0xAA : Chopper controller is not enabled | Chopper |
| A | DMC_MOVE_CHOP_REL_DITHER | 56 | INT32 | Move chopper relative + dither.<br>PARAM :<br>In closed loop :<br>Target position is in DMC_CHOP_CUR_POS units, i.e.<br>-32767 = -10V, 32767 = 10V<br>In open loop :<br>Target is the commanded current in the DAC and is therefore in DMC_CHOP_OUTPUT units, i.e.<br>-32767 = -133 mA, 32767 = 133 mA | Chopper |

| | | | | | |
|---|---|---|---|---|---|
| | | | | FAILURE CODE :<br>0xAA : Chopper controller is not enabled | |
| T | DMC_SET_CHOP_COIL_DRIV<br>E | 57 | UINT32 | Coil Drive Mode (bypass broken sections) + need to update PID parameters !!!!!! :<br>PARAM :<br>bits0-7 contain configuration of relays<br>- Normal Mode (0x28)<br>- Bypass coil 1 (0x30)<br>- Bypass coil 3 (0x48)<br>- Bypass coil 1 and coil 3 (0x50)<br>bits8 selects the open-loop mode :<br>- if bit8 is set to 1, open loop mode is active<br>- if bit8 is set to 0, closed loop mode is active<br>FAILURE CODE :<br>0xAA : Chopper controller is enabled | Chopper |
| T | DMC_SWON_FW_SPEC | 58 | NONE | Connect spectro FW to driving amplifier | FW |
| T | DMC_SWON_FW_PHOTO | 59 | NONE | Connect photo FW to driving amplifier | FW |
| T | DMC_SWON_BD_HEATER | 60 | NONE | Switch on Blue DEC heater<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SWOF_BD_HEATER | 61 | NONE | Switch off Blue DEC heater<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SWON_BD_FLASHER | 62 | NONE | Switch on Blue DEC flasher<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_SWOF_BD_FLASHER | 63 | NONE | Switch off Blue DEC flasher<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC blue |
| T | DMC_MOVE_SPEC_FW_LOC | 64 | UINT32 | Move FW Spec to Filter ID<br>PARAM:<br>0: Position A (0°)<br>1: Position B (180°)<br>2: Position A, opposite direction<br>3: Position B, opposite direction<br>DURATION : max 18.75 sec if rate is 100<br>FAILURE CODE :<br>0xAA : FW Spec is not powered on | FW |
| T | DMC_MOVE_SPEC_FW_STEP | 65 | INT32 | Move FW Spec by a number of steps<br>PARAM:<br>There are 6*256=1536 steps for 360°.<br>Allowed values for this param is [-1536, +1536]<br>DURATION : 18.75 sec for 360° if rate is 100<br>FAILURE CODE :<br>0xAA : FW Spec controller is not powered on | FW |
| T | DMC_MOVE_PHOTO_FW_LO<br>C | 66 | UINT32 | Move FW Photo to Filter ID<br>PARAM:<br>0: Position A (0°)<br>1: Position B (180°)<br>2: Position A, opposite direction<br>3: Position B, opposite direction<br>DURATION : max 18.75 sec if rate is 100<br>FAILURE CODE : | FW |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 0xAA : FW Spec is not powered on | |
| T | DMC_MOVE_PHOTO_FW_STEP | 67 | INT32 | Move FW Photo by a number of steps<br>PARAM:<br>There are 6*256=1536 steps for 360°.<br>Allowed values for this param is [-1536, +1536]<br>DURATION : 18.75 sec for 360° if rate is 100<br>FAILURE CODE :<br>0xAA : FW Spec controller is not powered on | FW |
| T | DMC_SWON_BB_1_CONT | 68 | NONE | Switch-on the BB1 controller => measure is valid<br>Start measurement in reading only mode | Calibration Sources |
| T | DMC_SWOF_BB_1_CONT | 69 | NONE | Switch-off the BB1 controller => measure is invalid | Calibration Sources |
| T | DMC_SET_TEMP_BB_1 | 70 | UINT32 | Modify the BB1 temperature setpoint.<br>PARAM:<br>The setpoint is the value of the resistor of the source. (1 unit = 100µohm) | Calibration Sources |
| T | DMC_SET_BB_1_VOLTAGE | 71 | INT32 | Directly set supply voltage in heating mode (controller must be disabled)<br>PARAM:<br>The voltage to apply to the BB1:<br>0 = 0V<br>32767 = +10V<br>NOTE :<br>Square wave voltage output is applied.<br>FAILURE CODE :<br>0xAA : BB1 is not powered on or controller is still enabled | Calibration Sources |
| T | DMC_SWON_BB_2_CONT | 72 | NONE | Switch-on the BB2 controller => measure is valid<br>Start measurement in reading only mode | Calibration Sources |
| T | DMC_SWOF_BB_2_CONT | 73 | NONE | Switch-off the BB2 controller => measure is invalid | Calibration Sources |
| T | DMC_SET_TEMP_BB_2 | 74 | UINT32 | Modify the BB2 temperature setpoint.<br>PARAM:<br>The setpoint is the value of the resistor of the source. (1 unit = 100µohm) | Calibration Sources |
| T | DMC_SET_BB_2_VOLTAGE | 75 | INT32 | Directly set supply voltage in heating mode (controller must be disabled)<br>PARAM:<br>The voltage to apply to the BB2:<br>0 = 0V<br>32767 = +10V<br>NOTE :<br>Square wave voltage output is applied.<br>FAILURE CODE :<br>0xAA : BB2 is not powered on or controller is still enabled | Calibration Sources |
| T | DMC_START_DIAG_HK | 76 | UINT32 | Start diagnostic HK.<br>PARAM:<br>the parameter is the period (in ms) between two housekeeping acquisitions with the following special values :<br>0 = 1 KHz<br>1 = Synchronize on blue spectrometer readouts<br>2 = Synchronize on red spectrometer readouts<br>4 = Synchronize on BOL readouts<br>Maximum value is 65535.<br>NOTE :<br>You must be very careful when using the | HK Diagnostic |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 1KHz mode, 15 measures acquired at 1KHz consume 20% of CPU workload which is not acceptable if the CPU is already at full load (Both DECs connected, transmitting to both SPUs and all mechanisms enabled)<br>FAILURE CODE :<br>0xAA : HK acquisition already running | |
| T | DMC_STOP_DIAG_HK | 77 | NONE | Stops diagnostic HK at the end of current interval. | HK Diagnostic |
| I | DMC_START_HK | 78 | NONE | Internal command sent at start-up to start the hk acquisition. It validates the core of the OBS | |
| A | DMC_SWON_RD_HEATER | 79 | NONE | Switch on Red DEC heater<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| A | DMC_SWOF_RD_HEATER | 80 | NONE | Switch off Red DEC heater<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| A | DMC_SWON_RD_FLASHER | 81 | NONE | Switch on Red DEC flasher<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| A | DMC_SWOF_RD_FLASHER | 82 | NONE | Switch off Red DEC flasher<br>FAILURE CODE :<br>0xAA : DEC is not switched-on or the connection between DMC and DEC is not established | DEC red |
| I | DMC_FW_GR_DAC_OUT | 83 | UINT32 | Directly writes in the FW and Grating DAC. Internal command to be used on ground only with no mechanism connected !<br>PARAM:<br>16msb: DAC1<br>16lsb: DAC2<br>32767 = 550mA | |
| I | DMC_SPARE_CMD_3 | 84 | NONE | Spare Command<br>FAILURE CODE :<br>0xA7 : Invalid trigger ID | |
| I | DMC_SEND_COMMAND_TO_ BLUE_DEC | 85 | UINT32 | command to blue DEC | |
| T | DMC_START_RED_SPU_LINK | 86 | UINT32 | 0 = Slave<br>1 = master | 1355 |
| T | DMC_START_BLUE_SPU_LIN K | 87 | UINT32 | 0 = Slave<br>1 = master | 1355 |
| T | DMC_COPY_OBS_TO_EEPRO M | 88 | NONE | Copy the OBS in EEPROM.<br>DURATION : 10 seconds. Avoid sending other commands during the writing. | OBSW |
| T | DMC_RESET_SMCS_CHIP_2 | 89 | NONE | Reset the SMCS2. | 1355 |
| T | DMC_SELECT_MECH_CTRL_ MODE | 90 | UINT32 | Select in which mode each of the mechanism controller will be used<br>PARAM (bit field):<br>bit0:<br>0 = grating use real position<br>1 = grating is simulated (pos = setpoint)<br>bit1:<br>0 = chopper controller use real position<br>1 = chopper pos is simulated (pos = setpoint)<br>bit2: | Mechanisms |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 0 = fw_spec use nominal position<br>1 = fw_spec use simulated position<br>Bit3:<br>0 = fw_photo use nominal position<br>1 = fw_photo use simulated position<br>Bit4:<br>0 = CS1 use nominal resistor reading<br>1 = CS1 is simulated (resistor = setpoint)<br>Bit5:<br>0 = CS2 use nominal resistor reading<br>1 = CS2 is simulated (resistor = setpoint)<br>Bit6 :<br>0 = nominal FS is execute<br>1 = FPGA status register diagnostic mode | |
| T | DMC_ENABLE_BB_1_CONT | 91 | NONE | Enable BB1 Controller | Calibration Sources |
| T | DMC_DISABLE_BB_1_CONT | 92 | NONE | Disable BB1 Controller | Calibration Sources |
| T | DMC_ENABLE_BB_2_CONT | 93 | NONE | Enable BB2 Controller | Calibration Sources |
| T | DMC_DISABLE_BB_2_CONT | 94 | NONE | Disable BB2 Controller | Calibration Sources |
| T | DMC_SWON_TEMP_SENSOR S | 95 | NONE | Switch-on temperature sensors in FPU: Chopper, BB1, FPU1, FPU2, Grating, FWSpec, FWPhoto | Temperature Sensors |
| T | DMC_SWOF_TEMP_SENSOR S | 96 | NONE | Switch-off temperature sensors in FPU | Temperature Sensors |

## 4.2 Write Commands

Below you will find a list of the write commands. Their format is described in [AD6].

### 4.2.1 Parameters arrays formats

| Activity ID | Function name | Description | Param ID | Length (words) | Subsystem |
|---|---|---|---|---|---|
| 128 | DMC_WRT_TIME | This buffer contains the time sent from DPU to DEC/MEC. Only 6 bytes are actually used (the first word contains the seconds, the 2 LSB of the second words contain the 1/65535th sec). Note : The change of Time occurs only when the DMC_SET_TIME command is called. At startup, the buffer is initialized with zeros. | 0 | 2 | Time Stamping |
| 129 | DMC_WRT_SEQ_BUFFER | Sequence Buffer : This buffer can contains maximum 256 commands stored on 2 words : the activity ID and the parameter. Note : The last command of the sequence must always be DMC_END_SEQUENCE. At startup, this buffer is filled by a default sequence (TBC). | 1 | max 512 | Sequencer |
| 130 | DMC_WRT_SEQ_BUFFER_0 | Sequence Buffer Split 0 : For some memory write commands (the ones coming from ground), the size is limited to 214 bytes. So, if we want to upload a big sequence, we have to split the command in a few write commands. For that purpose, we define ID's allowing us to | 2 | max 52 | Sequencer |

| | | | | | |
|---|---|---|---|---|---|
| | | access different subset of the sequence buffer.<br>Note : The Sequence Buffer Split Part 0 points to the same address as the Sequence Buffer. | | | |
| 131 | DMC_WRT_SEQ_BUFFER_1 | This buffer points to the elements 26 to 51 of the Sequence Buffer. | 3 | max 52 | Sequencer |
| 132 | DMC_WRT_SEQ_BUFFER_2 | This buffer points to the elements 52 to 77 of the Sequence Buffer. | 4 | max 52 | Sequencer |
| 133 | DMC_WRT_SEQ_BUFFER_3 | This buffer points to the elements 78 to 103 of the Sequence Buffer. | 5 | max 52 | Sequencer |
| 134 | DMC_WRT_SEQ_BUFFER_4 | This buffer points to the elements 104 to 129 of the Sequence Buffer. | 6 | max 52 | Sequencer |
| 135 | DMC_WRT_SEQ_BUFFER_5 | This buffer points to the elements 130 to 155 of the Sequence Buffer. | 7 | max 52 | Sequencer |
| 136 | DMC_WRT_SEQ_BUFFER_6 | This buffer points to the elements 156 to 181 of the Sequence Buffer. | 8 | max 52 | Sequencer |
| 137 | DMC_WRT_SEQ_BUFFER_7 | This buffer points to the elements 182 to 207 of the Sequence Buffer. | 9 | max 52 | Sequencer |
| 138 | DMC_WRT_SEQ_BUFFER_8 | This buffer points to the elements 208 to 233 of the Sequence Buffer. | 10 | max 52 | Sequencer |
| 139 | DMC_WRT_SEQ_BUFFER_9 | This buffer points to the elements 234 to 255 of the Sequence Buffer. | 11 | max 44 | Sequencer |
| 140 | DMC_WRT_NOT_USED_1 | Writing one word in this value will not have any influence on the execution but it will not generate any error message. | 12 | 1 | |
| 141 | DMC_WRT_DIAG_HK_LIST | Housekeeping Diagnostic list : This buffer contains the list of Ids of HK Measures that are requested in the housekeeping diagnostic packet. Each ID is stored in one word (although only 16 bits are useful). The last ID must always be END_OF_HK_LIST (0xFFFF). At startup, the list is empty. | 13 | max 16 | HK Diagnostic |
| 142 | DMC_WRT_DIAG_HK_CONF_TAB | Custom Hk Configuration Table : This buffer contains configuration data allowing us to increase the number of available Hk measure without recompiling the application. The table is composed of 10 entries each of them being 5 words long :<br>q the address of the memory area to monitor<br>q the size (in words) of the memory area to monitor<br>q the size (in bytes) of the memory area to monitor<br>q a pointer to a function performing the monitoring (if you want to use this field, it is highly probable that you need to patch your code anyway to upload the new monitoring function).<br>q the validity at startup (0 = invalid, 1 = valid)<br><br>All the fields are initialized to zero at startup. (for advanced users only) | 14 | max 50 | HK Diagnostic |
| 143 | DMC_WRT_GRAT_CONF_PAR | Parameters to configure the Grating servo loop.<br>Start-up value and parameters description is given below | 15 | 9 | Grating |
| 144 | DMC_WRT_CHOP_CONF_PAR | Parameters to configure the Chopper servo loop.<br>Start-up value and parameters description is | 16 | 21 | Chopper |

| | | | | | |
|---|---|---|---|---|---|
| | | given below. | | | |
| 145 | DMC_WRT_FW_SPEC_CONF_PAR | Parameters to configure the FW SPEC servo loop.<br>Start-up value and parameters description is given below. | 17 | 6 | FW |
| 146 | DMC_WRT_FW_PHOT_CONF_PAR | Parameters to configure the FW PHOTO servo loop.<br>Start-up value and parameters description is given below. | 18 | 6 | FW |
| 147 | DMC_WRT_CS1_CONF_PAR | Parameters to configure the CS1 Temperature regulator.<br>- Kp : Proportional gain of the PI regulator (default value = 1000000)<br>- Ki : Integral gain of the PI regulator (default value = 5000)<br>- Maximum Accumulator Limit : The max value of the PI regulator accumulator (default value = 3277)<br>- Minimum Accumulator Limit : The min value of the PI regulator accumulator (default value = 1857)<br>- Output Threshold : The minimum output of the regulator; below this limit, the regulator is in "measure only" mode and this value is used to make the measure (default value = 327)<br>- Output Limit : The maximum output of the regulator (default value = 0x7FFF)<br>- DAC offset : value added to the output to cancel the DAC offset (default value = 0) | 19 | 7 | Calibration Sources |
| 148 | DMC_WRT_CS2_CONF_PAR | Parameters to configure the CS2 Temperature regulator.<br>- Kp : Proportional gain of the PI regulator (default value = 1000000)<br>- Ki : Integral gain of the PI regulator (default value = 5000)<br>- Maximum Accumulator Limit : The max value of the PI regulator accumulator (default value = 3277)<br>- Minimum Accumulator Limit : The min value of the PI regulator accumulator (default value = 1857)<br>- Output Threshold : The minimum output of the regulator; below this limit, the regulator is in "measure only" mode and this value is used to make the measure (default value = 327)<br>- Output Limit : The maximum output of the regulator (default value = 0x7FFF)<br>- DAC offset : value added to the output to cancel the DAC offset (default value = 0) | 20 | 7 | Calibration Sources |
| 149 | DMC_WRT_GRAT_MAX_POWER | Output limit that will trigger the power limit error in case this limit is reached during 5s.<br>This value is also used to trigger the end of the homing (when this limit is reached during 0.8s)<br>Default values is 8855 (150mA) (conversion formula: 32767 = 555mA) | 21 | 1 | Grating |
| 150 | DMC_WRT_BOL_REC_OPT | BOL Receiver Options<br>Only the 8 LSB are used as a bit field.<br>At startup, its value is 0x04.<br>The following options are defined : | 22 | 1 | 1355 |

| | | | | | |
|---|---|---|---|---|---|
| | | bit2 : 0 = forward data to Packet Encoder, 1 = don't forward data to Packet Encoder (data are lost) | | | |
| 151 | DMC_WRT_B_DEC_REC_OPT | Blue DEC Receiver Options<br>Only the 8 LSB are used as a bit field.<br>At startup, its value is 0x04.<br>The following options are defined :<br>bit2 : 0 = forward data to Packet Encoder, 1 = don't forward data to Packet Encoder (data are lost) | 23 | 1 | 1355 |
| 152 | DMC_WRT_R_DEC_REC_OPT | Red DEC Receiver Options<br>Only the 8 LSB are used as a bit field.<br>At startup, its value is 0x04.<br>The following options are defined :<br>bit2 : 0 = forward data to Packet Encoder, 1 = don't forward data to Packet Encoder (data are lost) | 24 | 1 | 1355 |
| 153 | DMC_WRT_MAX_DITHER | Dithering Amplitude : This integer contains the maximum dithering amplitude (in encoder steps).<br>At startup, Dithering Amplitude = 16 (TBC). | 25 | 1 | Chopper |
| 154 | DMC_WRT_R_SPEC_PAR | Red DEC Parameters : This buffer contains the set of parameters that will be uploaded to red DEC by the SET_PARAM_RED_SPEC_ARRAY command.<br>Start-up value is given below.<br>See format description below | 26 | 6 | DEC red |
| 155 | DMC_WRT_B_SPEC_PAR | Blue DEC Parameters : This buffer contains the set of parameters that will be uploaded to blue DEC by the SET_PARAM_BLUE_SPEC_ARRAY command.<br>Start-up value is given below.<br>See format description below | 27 | 6 | DEC blue |
| 156 | DMC_WRT_SPU_TRAN_MODE | SPU Transmission Modes :<br>- Blue SPU transmission mode (default = 0x10)<br>- Red SPU transmission mode (default = 0x10)<br>These words are inserted in the packets sent to the SPU. | 28 | 2 | 1355 |
| 157 | DMC_WRT_TIMING_FPGA_PAR | See below | 29 | max 6 | Synchro |
| 158 | DMC_WRT_B_PACKT_ENC_LINK | ID of the link on which blue science data must be output<br>- output to blue SPU (default) : 2<br>- output to red SPU : 3 (on AVM : 5) | 30 | 1 | 1355 |
| 159 | DMC_WRT_R_PACKT_ENC_LINK | ID of the link on which red science data must be output<br>- output to blue SPU : 2<br>- output to red SPU (default) : 3 (on AVM : 5) | 31 | 1 | 1355 |
| 160 | DMC_WRT_GRAT_INDUCT_AMPL | Amplitude of the inductosyn excitation signal (0 = 0V, 4096 = 5V) | 32 | 1 | Grating |
| 161 | DMC_WRT_GRAT_RANGE | This parameter contains the number of encoder steps from one hardstop to the other. See in the section 'Using the grating' to see how to determine it. In the software, the default value is 0xF0000.<br>NOTE : The grating PFM has a range of | 33 | 1 | Grating |

| | | | | | |
|---|---|---|---|---|---|
| | | 0x100000. | | | |
| 162 | DMC_WRT_GRAT_HALL_OFFSET | Parameter used to offset the hall sensors of the grating such that their values are centered on zero.Default value is 3000. | 34 | 1 | Grating |
| 163 | DMC_WRT_GRAT_DEG_MODE_PARAM | Parameters used for the grating degraded mode.<br>- Rate : number of execution of the ISR between two steps in the sine table (1 step = 13,18 arcsec) (default value = 32 => 256 steps per second)<br>- Maximum Output Current (default value = 8192 = 138,75mA), (conversion formula: 32767 = 555mA ) | 35 | 2 | Grating |
| 164 | DMC_WRT_GRAT_CONF_FILT | Parameters used for the filter applied on the output of the grating controller.<br>The parameter block is composed of 5 coefficients (N1, N2, N3, D1, D2).<br>They must be entered as integer values and are converted to float values at the time of the grating enable command. There is a 1000000 ratio between integer values and float values (1000000 in integer gives 1.0 in float).<br>Default values:<br>N1 = 1000000<br>N2, N3, D1, D2 = 0 | 36 | 5 | Grating |

#### 4.2.1.1 Red DEC parameters and Blue DEC parameters

The table below contains the description of the parameter array that can be written in DMC_WRT_B_SPEC_PAR and DMC_WRT_R_SPEC_PAR.

This array defines the values for the OBSW 5.016 and later and the DEC FPGAs delivered after the first DMC EM.

| word | Parameter name | Range of values | Hardware, at power up | Initial values in data block |
|---|---|---|---|---|
| 0 | Clocks_per_readout | 32-255 | 32 | 32 |
| 1 | Readouts_per_ramp | 2-65535 | 8 | 8 |
| 2 | CRE_ctrl_reg | Bit field (see below) | 8Ch | 8Ch |
| 3 | Bias_r_command (bias applied to the resistor (dummy detector)) | 0-4095 (4095 = 1V) | 0 | 0 |
| 4 | Bias_d_command (bias applied to the detector) | 0-4095 (4095 = 1V) | 0 | 0 |
| 5 | Simul_reg | Any (16 bits) | EA60h | EA60h |

The table below contains a description of the CRE_Ctrl_Reg bit field.

| Bit position | name | Active state ( logic 1 ) function | Remarks |
|---|---|---|---|
| 0 (lsb) | Power on (don't care) | Hardware applies bias and supplies to CRE.<br>Note: This bit is used by the trigger commands that switch on/off the detector arrays. So, the value of this bit written by the write command is not considered. Its description is given for information purpose only (it is accessible in the HK). | Hardware manages order of applying voltages |
| 1 | enable | CRE logic enabled and current sources active | Connects to CRE line "select" and to output lines bias current sources in DEC front end |
| 2 | SEL1 | Capacitor select line SEL1 | Name and number changed with QM CRE ICD, corresponding capacitor values are listed below. |
| 3 | SEL2 | Capacitor select line SEL2 | |
| 4 | curing | Puts electronics and CRE in curing mode | Activate CRE SYNC to force reset state + apply bias to detector pixels |
| 5 | Spare | Spare | Spare |
| 6 | simulation | Hardware put in simulation mode | Enable pixel data simulation (see section 4.4.2) |
| 7 | T° sensors control | T° sensors are biased | Apply bias current to 2K and 4K sensors |
| 8 | Spare | Spare | Spare |
| 9 | Ramp simulation | Hardware simulating ramps | Enable ramp simulation<br>(see section 4.4.2) |
| 10 | Switch on/off heater (don't care) | Note: This bit is used by the trigger commands that switch on/off the heater. So, the value of this bit written by the write command is not considered. Its description is given for information purpose only (it is accessible in the HK). | 0=OFF<br>1=ON |
| 11 | Switch on/off flasher (don't care) | Note: This bit is used by the trigger commands that switch on/off the flasher. So, the value of this bit written by the write command is not considered. Its description is given for information purpose only (it is accessible in the HK). | 0=OFF<br>1=ON |

The table below contains a description of values that can be given to bit 2 and 3 of CRE_CTRL_REG to select the capacitors.

| Bit3 SEL2 | Bit2 SEL1 | Integrating capacitor value |
|---|---|---|
| 0 | 0 | 100 fF |
| 1 | 0 | 200 fF |
| 0 | 1 | 400 fF |
| 1 | 1 | 1 pF |

#### 4.2.1.2 Grating parameters

The grating controller is a PID controller using the following function :

$$Iout = \frac{Kp}{1000}.\varepsilon_t + \frac{Ki}{1000.F}.\sum_t \varepsilon + \frac{Kd}{1000}.F.(\varepsilon_t - \varepsilon_{t-1})$$

With :

Iout = DMC_GRAT_OUTPUT

ε = DMC_GRAT_PID_ERROR

F = Interrupt routine frequency (nominally 8192 Hz)

$\Sigma_t$ ε = DMC_GRAT_PID_ACC

The table below contains a description of the parameter array that can be written in *DMC_WRT_GRAT_CONF_PAR*

| word | type | name | Remarks | Default values in OBSW |
|---|---|---|---|---|
| 0 | int | Kp | Proportional gain | 0x3e8 |
| 1 | int | Ki | Integral gain | 0xc350 |
| 2 | int | Kd | Differential gain | 0x12 |
| 3 | int | Filter order | The order of the filtering applied on the speed of the grating. 0 means no filter. | 0x0 |
| 4 | int | Rate | Setpoint increment at each execution of the PID controller (each ISR execution). If rate = 3, grating will move by 3*8192=24576 unit/sec (around 1°/sec). The setpoint is incremented using the rate parameter until the commanded target value is reached. Note that the rate must not be higher than 3 for homing operations but can be increased to higher values otherwise. Tests have been performed on grating PFM with a rate of 12 (4°/sec.) successfully. | 0x3 |
| 5 | int | Accumulator Limit | PID controller accumulator limit. Above this value, the accumulator (DMC_GRAT_PID_ACC) will not be updated anymore. This can be used to avoid large overshoot. Attention : the value must never be higher than 0x7fffffff-(MaxError*Ki) where MaxError is the maximum error ever expected. | 0x51eae1 |
| 6 | int | OutputLimit | PID controller output limit, maximum commanded current (32767 = 555mA). If the output is equal to this limit during 1 sec, the controller will be disabled (see below: Power Limit Error) | 0x5c3f |
| 7 | int | Scaling | Hall sensor amplitude scaling. This is used to scale the amplitude of the actuator hall sensors to get an amplitude of 65536. A value of 1024 results in a scaling of 1. See §4.4.8.1 for more details. | 0x6E5 |
| 8 | int | ErrorLimit | Maximum error. If the error gets bigger than this limit, the controller will be disabled (see below: Error Limit Error). | 0x5b06 |

The *DMC_WRT_GRAT_HALL_OFFSET* must also be changed for each model/temperature. This parameter is used to correct the offset of the actuator hall sensors. See §4.4.8.1 for more details.

The *DMC_WRT_GRAT_INDUCT_AMPL* must also be changed for each model/temperature. This parameter is used to adjust the maximum amplitude of the inductosyn sine and cosine signals to 2±0.2 Vrms. See §4.4.8.1 for more details.

The *DMC_WRT_GRAT_RANGE* must be used to set the range of the used grating. Note that this is not depending of temperature.

The DMC_WRT_GRAT_CONF_FILT must be used to change the output filter. This function is writing integer coefficients of the filter that must be converted into float values. The conversion from integer to flot is done in the DMC_ENABLE_GRAT_CONT.

The parameters used for the PFM grating during the acceptance test campaign at CSL are given in the table below (TBC). See PACS-CL-TR-019 and PACS-CL-TR-021 for test reports.

| **Parameter** | **Nominal circuit** | | | **Redundant circuit** | | |
|---|---|---|---|---|---|---|
| | Ambient | 4.2 K | | Ambient | 4.2 K | |
| | | Normal | Freq. switch | | Normal | Freq. switch |
| Kp | 0x38e | 0x1388 | 0x32c8 | 0x38e | 0x1388 | 0x2af8 |
| Ki | 0x7530 | 0x3d090 | 0xc350 | 0x7530 | 0x3d090 | 0xc350 |
| Kd | 0x12 | 0x28 | 0x28 | 0x12 | 0x28 | 0x28 |
| Kf | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 |
| Rate | 0x3 | 0x3 | 0x5 | 0x3 | 0x3 | 0x5 |
| AccumulatorLimit | 0x10e4311 | 0x10e4311 | 0x10e4311 | 0x10e4311 | 0x10e4311 | 0x10e4311 |
| OutputLimit | 0x452f | 0x452f | 0x452f | 0x452f | 0x452f | 0x452f |
| Scaling | 0x8a4 | 0x71c | 0x71c | 0x859 | 0x6c9 | 0x6c9 |
| ErrorLimit | 0x5b06 | 0x5b06 | 0x5b06 | 0x5b06 | 0x5b06 | 0x5b06 |
| Hall sensor offset | 0x17d | 0xd53 | 0xd53 | -0x93 | -0xd8b | -0xd8b |
| Inductosyn ampl. | TBD | TBD | TBD | TBD | TBD | TBD |
| Grating range | 0x100000 | 0x100000 | 0x100000 | 0x100000 | 0x100000 | 0x100000 |

#### 4.2.1.3 Filter wheel parameters

The filter wheels are controlled in open loop as explained in §4.4.11.
The table below contains a description of the parameter array that can be written in *DMC_WRT_FW_SPEC_CONF_PAR* or *DMC_WRT_FW_PHOT_CONF_PAR*

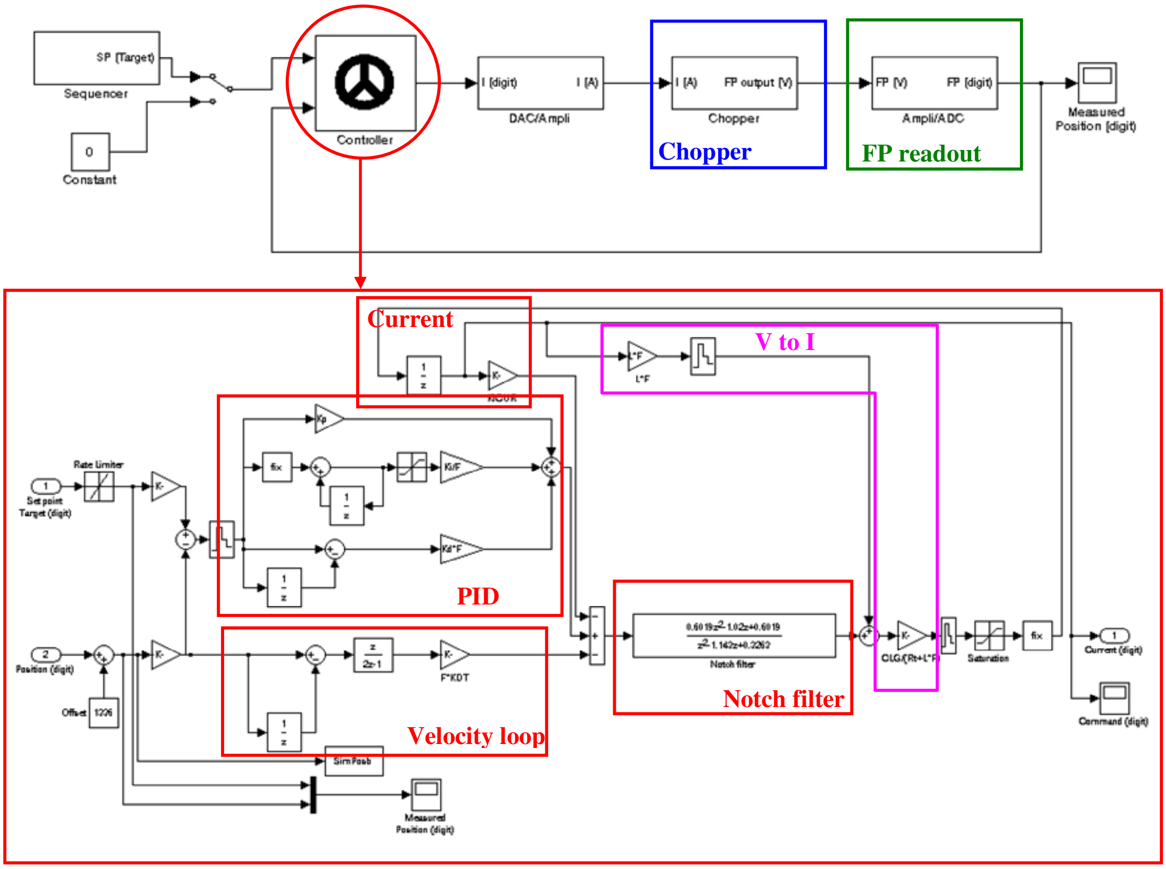
| **word** | **type** | **name** | **remarks** | **Default values in OBSW** |
|---|---|---|---|---|
| 0 | int | Rate | The number of interrupt count between two steps in the sine table (default = 100 => 100*1536/8192=18.75 sec for 360°). This will define the frequency of the sine and cosine driving | 0x64 |

| | | | | |
|---|---|---|---|---|
| | | | functions sent to the coils of the actuator and therefore determine the speed. | |
| 1 | int | Maximum Output Current | Default = 4096 = 69.4mA.<br>Max value supported by the amplifier at ambient T° = 4133 = 70mA due to high coils impedance; max value at cold temperature = 32767 = 555mA). This actually defines the amplitude of the sine and cosine driving functions sent to the coils of the actuator. For example, if parameter = 4133, sine and cosine of +/-70 mA amplitude will be generated. This determine the torque of the actuator and therefore the acceleration. | 0x1000 |
| 2 | int | Switch A Control Threshold | Conversion formula: 32767 = 10V. Once the Hall Sensor of the position sensors reaches this limit, the FW is considered to be close to position A and no current is sent in the coils anymore. (flagged in bit 30 of the controller status word) | Photo = 6725<br>Spectro = 8365 |
| 3 | int | Switch B Control Threshold | Conversion formula: 32767 = 10V. Once the Hall Sensor of the position sensors reaches this limit, the FW is considered to be close to position B and no current is sent in the coils anymore. (flagged in bit 31 of the controller status word) | Photo = 7830<br>Spectro = 6175 |
| 4 | int | Switch A Status Threshold | Conversion formula: 32767 = 10V. Once the Hall Sensor of the position sensors is above this limit, the 'position A' bit will be set in the controller status word of the FW (flagged in bit 28 of the controller status word) | 0x7d0 |
| 5 | int | Switch B Status Threshold | Conversion formula: 32767 = 10V. Once the Hall Sensor of the position sensors is above this limit, the 'position B' bit will be set in the controller status word of the FW (flagged in bit 29 of the controller status word) | 0x7d0 |

Maximum Output Current and Rate parameters must be adjusted carefully together in order to avoid high oscillation of the wheel during a move.

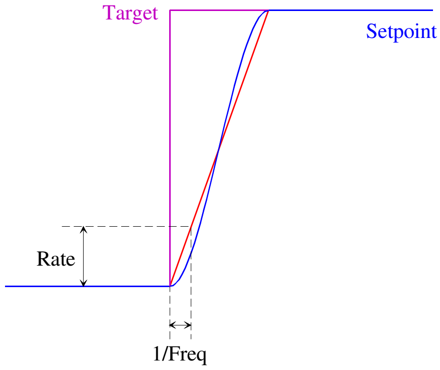
#### 4.2.1.4 Chopper parameters

The chopper controller has been defined by Zeiss (refer to PACS-MA-TN-678 for more details) and has the architecture as defined in the figure below. The controller is mainly composed of a PID control acting together with a velocity loop and a current loop feedbacks. Finally, the output of the controller is filtered through a digital filter to damp the resonnance frequencies of the chopper structure and rotor. Furthermore, as Zeiss defined a voltage output controller while the DMC electronics specification was to control mechanism with current output amplifiers, a voltage to current conversion is made.




The table below contains a description of the parameter array that can be written in *DMC_WRT_CHOP_CONF_PAR*

| word | type | name | remarks | Default values in OBSW |
|---|---|---|---|---|
| 0 | int | Kp | Proportional gain (see below for definition) | 0x9F600 |
| 1 | int | Ki | Integral gain (see below for definition) | 0x4618560 |
| 2 | int | Kd | Differential gain (see below for definition) | 0x270 |
| 3 | int | Kf | Velocity loop gain (see below for definition) | 0x9C4 |
| 4 | int | Rate | Determine how Setpoint is incremented at each execution of the PID controller (each ISR execution).<br>This parameter determines the speed of the chopper during transition and therefore the transition time.<br>The rate can be computed using the following formula :<br>Rate = Amplitude / (Ttrans * Freq) where<br>Amplitude = movement amplitude in setpoint units<br>Ttrans = transition time in seconds<br>Freq = sampling frequency (= interrupt routine frequency, usually 8192 Hz)<br>For example, chopping with an amplitude of 26000 at a chopping frequency of 10 Hz with a transition time of 10 ms (20%) will be done using Rate = 26000/(0.01*8192) = 317. | 0x148 |

| | | | | |
|---|---|---|---|---|
| | | | The actual setpoint is incremented using a sine function related to the Rate parameter as shown in the figure below.<br> | |
| 5 | int | Accumulator Limit | PID controller accumulator limit. Above this value, the accumulator (DMC_CHOP_PID_ACC) will not be updated anymore. This can be used to avoid large overshoot.<br>Attention : the value must never be higher than 0x7fffffff-MaxError where MaxError is the maximum error ever expected (0xFFFF). | 0x3fffffff |
| 6 | int | OutputLimit | PID controller output limit, maximum commanded current (32767 = 133mA). **<u>This value must be carefully adjusted for different coils configurations used.</u>**<br>In order to avoid any damage of the chopper, it is recommended to limit the output at 43 mA (0x2962) in nominal operation (3 coils) and increase the limit to 86 mA (0x52c4) (TBC) for 2 coils operation and to 133 mA (0x7fff) for 1 coil operation. Actually, the 3 coils are not identicals (2 small + 1 big) and therefore, consistent limits must be determined by test. See PACS-MA-TN-678. | 0x7fff |
| 7 | int | PosLimit | Position limit. If the chopper goes further than this position, the controller is disabled and the ERR_CHOPPER_CONTROLLER_POSITION_ERROR is signalled. If it is set to 0x7FFFFFFF, it disables the position limit detection | 0x7FFFFFFF |
| 8 | int | ErrorLimit | If the error is bigger than this value, the 'following error' is signalled (but no autonomous action is taken). If it is set to 0x7FFFFFFF, it disables the following error detection. | 0x7FFFFFFF |
| 9 | Int | PosOffset | Offset added to the position readout to get 0 at the mechanical rest position of the chopper. Offset must be determined by recording the position of the chopper with no driving current. | 0x819 |
| 10 | Int | KiCurr | Current loop gain (see below for definition) | 0x668a0 |
| 11 | int | SelectFieldPlateLUT | Select the field plate lookup-table used to linearize the field plate output:<br>0 = FM nominal field plate (to use with nominal FM chopper)<br>1 = FM redundant field plate (to use with redundant FM chopper)<br>2 = FS nominal field plate (to use with nominal FS chopper)<br>3 = FS redundant field plate (to use with redundant FS chopper)<br>Other values = no lookup-table is used (same as with version 6.017 and before) | 0x0 |
| 12 | Int | Spare | Not used anymore | 0x0 |
| 13 | Int | FilterN1 | Filter factor (see below for definition) | 0x21F |
| 14 | Int | FilterN2 | Filter factor (see below for definition) | 0x345 |

| | | | | |
|---|---|---|---|---|
| 15 | Int | FilterN3 | Filter factor (see below for definition) | 0x21F |
| 16 | Int | FilterD1 | Filter factor (see below for definition) | 0x3EB |
| 17 | Int | FilterD2 | Filter factor (see below for definition) | 0xFB |
| 18 | Int | Inductance | Chopper inductance used in V to I conversion.<br>Units are in mH. | 0x97 |
| 19 | Int | Resistance | Chopper resistance used in V to I conversion.<br>Units are in mΩ. | 0x3C1E |
| 20 | Int | ControlLoop Gain | Control loop gain (see below for definition) | 0x3E8 |

The chopper controller parameters are usually defined by Zeiss and can be converted to DMC units as shown hereunder. The parameter conversion is each time given as example for the FM chopper for the case at cold temperature (8K), 3 coils configuration, 4.1 degrees, 10 Hz rectangle (see PACS-MA-TN-678 §5.1).

– Kp : proportional gain

Obtained using Zeiss parameters **KP**, **FPMult**, **FPMPS/FPMNS** (usually identical, if not take (FPMPS+FPMNS)/2) by the following formula :

Kp = KP*75180*FPMult*(FPMPS+FPMNS)/2*34.35/50

Example : Kp = 389431 (0x5f137)

– Ki : integral gain

Obtained using Zeiss parameters **KI**, **FPMult**, **FPMPS/FPMNS** (usually identical, if not take (FPMPS+FPMNS)/2) by the following formula :

Ki = KI*75180*FPMult*(FPMPS+FPMNS)/2*34.35/50

Example : Ki = 36696373 (0x22ff135)

– Kd : differential gain

Obtained using Zeiss parameters **KD**, **FPMult**, **FPMPS/FPMNS** (usually identical, if not take (FPMPS+FPMNS)/2) by the following formula :

Kd = KD*75180*FPMult*(FPMPS+FPMNS)/2*34.35/50

Example : Kd = 614 (0x266)

– Kf : velocity loop gain

Obtained using Zeiss parameters **KDT**, **FPMult**, **FPMPS/FPMNS** (usually identical, if not take (FPMPS+FPMNS)/2) by the following formula :

Kf = KDT*75180*FPMult*(FPMPS+FPMNS)/2*34.35/50

Example : Kf = 1198 (0x4ae)

– KiCurr : current loop gain

Obtained using Zeiss parameters **KICUR** by the following formula :

KiCurr = KICUR*100000
Example : KiCurr = 237000 (0x39dc8)

– <u>FilterN1, FilterN2, FilterN3, FilterD1, FilterD2</u>

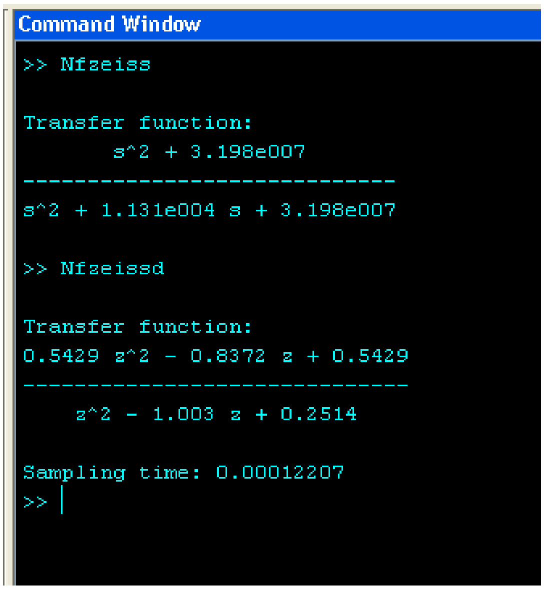
These parameters are defining the output filter. Any kind of filter can be used as soon as it can be defined based on the formula :

$$Y = \frac{\frac{FilterN1}{1e6}.z^2 - \frac{FilterN2}{1e6}.z + \frac{FilterN3}{1e6}}{z^2 - \frac{FilterD1}{1e6}.z + \frac{FilterD2}{1e6}}.X$$

Example 1: Notch filter

Obtained using Zeiss Notch filter parameters **F1, DA** and by discretisation of the Zeiss Notch filter function using the Matlab "c2d" function as following :

```
Nfzeiss=tf([1 0 (2*pi*F1)^2],[1 DA*2*pi*F1 (2*pi*F1)^2]);
Nfzeissd=c2d(Nfzeiss,1/8192,'matched');
```

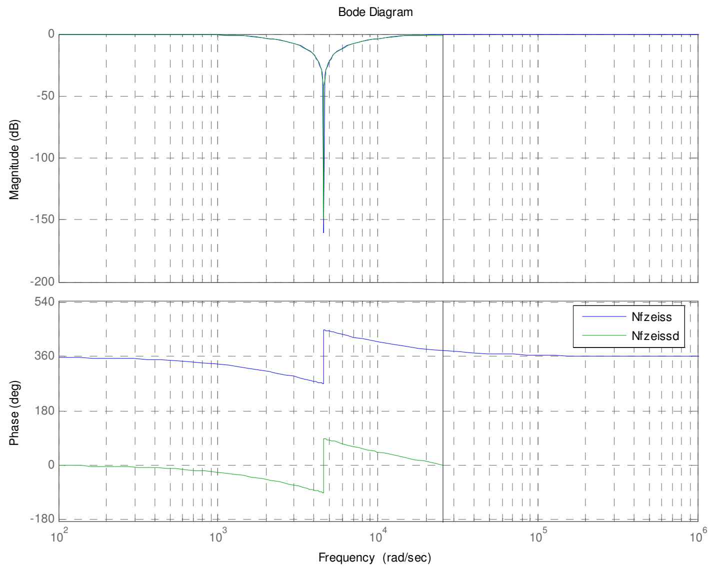



Therefore :

FilterN1 = 542900 (0x848b4)
FilterN2 = 837200 (0xcc650)
FilterN3 = 542900 (0x848b4)

FilterD1 = 1003000 (0xf4df8)
FilterD2 = 251400 (0x3d608)

The figure below shows the two functions Nfzeiss and Nfzeissd in a Bode diagram. Validity of the discretisation can be assessed.




Example 2 : Elliptic low pass filter

An elliptic filter of order N can be defined using the Matlab "ellip" function as following :

```
[N,D]=ellip(N,Rp,Rs,Wn);
lpfilter=tf(N,D,1/8192);
```

where

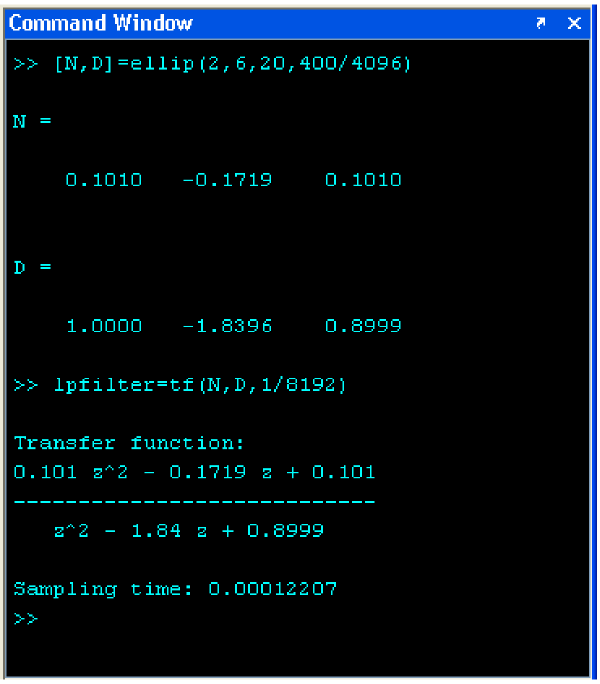
Rp = maximum ripple amplitude allowed in passband (in dB)

Rs = minimum attenuation in stopband (in dB)

Wn = normalised cutoff frequency (= cutoff frequency (in Hz) divided by half of the sampling frequency (= 4096))

The example is using :

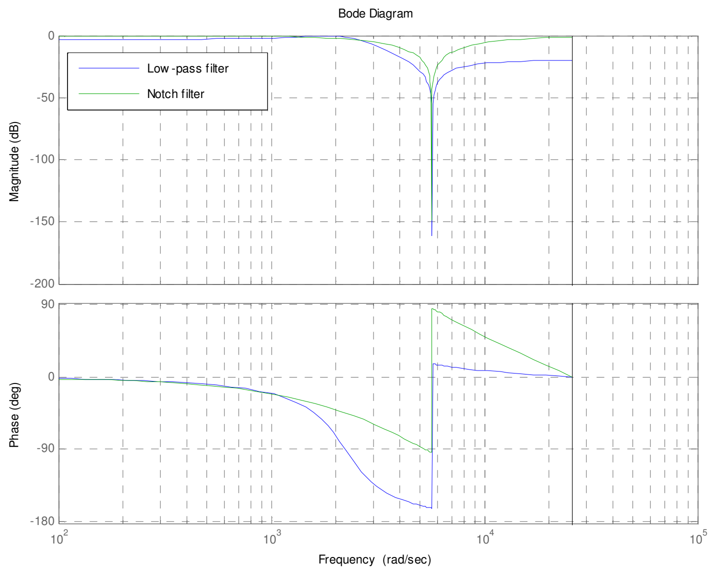
N=2, Rp=3, Rs=20, Wn=400/4096




Therefore :

FilterN1 = 101880 (0x18df8)
FilterN2 = 156500 (0x26354)
FilterN3 = 101880 (0x18df8)
FilterD1 = 1768200 (0x1afb08)
FilterD2 = 834956 (0xcbd8c)

The figure below shows the low pass filter lpfilter compared with the notch filter Nfzeissd (as defined in example 1) in a Bode diagram.




– <u>ControlLoopGain : control loop gain</u>

Obtained using Zeiss parameters **CLG** by the following formula :

ControlLoopGain = CLG*1000

Example : ControlLoopGain = 1000 (0x3e8)

#### 4.2.1.5 Calibration source parameters

The calibration source controller is a PI controller using the following function :

$$Vout = Kp.\varepsilon_t + \frac{Ki}{F}.\sum_t \varepsilon$$

With :

Vout = DMC_CS1_OUTPUT / DMC_CS2_OUTPUT

ε = (DMC_CS1_TARGET – DMC_CS1_RES_VALUE)/1000000

(DMC_CS2_TARGET – DMC_CS2_RES_VALUE)/1000000

F = Calibration source controller frequency (0.05 Hz)

$\Sigma_t \varepsilon$ / F = Calibration source accumulator (no HK variable for it)

The table below contains a description of the parameter array that can be written in *DMC_WRT_CS1_CONF_PAR / DMC_WRT_CS2_CONF_PAR.*

| word | type | name | remarks | Default values in OBSW |
|---|---|---|---|---|
| 0 | int | Kp | Proportional gain | 0xf4240 |
| 1 | int | Ki | Integral gain | 0x1388 |
| 2 | int | Maximum Accumulator Limit | Upper limitation to the controller accumulator. This is used to avoid large overshoot when heating the source for a long time. The integral part of the controller should be acting only when the setpoint is almost reached. Value is multiplied by 1000 to increase resolution. Division by 1000 is done by OBSW before use. | 0xccd |
| 3 | int | Minimum Accumulator Limit | Lower limitation to the controller accumulator. This is used to avoid large undershoot when cooling the source for a long time. The integral part of the controller should be acting only when the setpoint is almost reached. Value is multiplied by 1000 to increase resolution. Division by 1000 is done by OBSW before use. | 0x741 |
| 4 | int | Output Threshold | The minimum output of the regulator; below this limit, the regulator switch to the "reading only" mode and this value is used for the measurement step. See §4.4.13 for details | 0x147 |
| 5 | int | OutputLimit | PID controller output limit, maximum commanded voltage.<br>Currently not used. | 0x7fff |
| 6 | int | DAC Offset | Value added to the output to cancel an eventual DAC offset.<br>Currently not used. | 0x0 |

#### 4.2.1.6 Timing FPGA parameters

The table below contains a description of the parameter array that can be written in DMC_WRT_TIMING_FPGA_PAR.

| word | Parameter name | Parameter function | Range of values | Hardware, at power up | Initial values in data block (spectro) | Suggested values for photo mode |
|---|---|---|---|---|---|---|
| 0 | Sync_src_sel_reg | Obsolete: Synchronisation source selector (this parameter is now accessible only through the DMC_SYNCHRONIZE_ON_DET trigger command). The value is not used. | NA | 0 | 0 | 0 |
| 1 | Phase_shift_reg | Delay between external sync edge and software signal triggering the move of the mechanism (expressed in number of interrupt routine)<br>Note: when phase_shift_reg != 0, the actual delay is actually given by phase_shift_reg+1 (it is therefore impossible to set an actual phase shift | 0 to number of ISR per readout -1 (207 for photo, 31 for spectro) | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | of 1) | | | | |
| 2 | Bolc_freq_div | BOLC frequency divider (for debug/advanced users only) | 0 to 31 | 0 | 0 | 0 |
| 3 | Pd5_reg | Programmable divider 5 (for debug/advanced users only) | 0 to 511 | 26 | 26 | 26 |
| 4 | Phase_inc | Phase increment | See below | 0x095217CB | 0x095217CB | 0x0977602A |
| 5 | Control_register | Control register | See below | 0xB | 0xB | 0x22 and 0x32 (see section 4.4.3) |

The Phase_inc parameter shall only contain one of the recommended values given for each mode. Entering an invalid value (like 0 or something bigger than 0x7FFFFFFF) might completely block the DMC.

The table below contains the description of all the bits of Control_register. Note that bits 2, 22, 23, 24 are modified by DMC OBS and are therefore not modifiable by DPU command.

| **Bit (lsb=0)** | **ID** | **Power on value** | **When bit = 0** | **When bit = 1** |
|---|---|---|---|---|
| 0 | Not used | 0 | NA | |
| 1 | PERIOD_COUNT_ENB<br>Period count enable: enable period measurement. The period is the interval between 2 synchro signals (received from DECs or BOLC). When enabled, the DMC_OBT_COUNT is updated every time a synchro signal is received. | 0 | Period measurement circuit disabled | Period measurement circuit enabled |
| 2 | Not used | 0 | NA | |
| 3 | RESET_PLL_BY_OBT<br>When enabled, the PLL locks on the OBT. Otherwise, the PLL is in free run. | 0 | OBT is not connected to PLL. PLL is free running | PLL is locked on OBT. (for SPECTRO mode) |
| 4 | PLL_PD_SYNC<br>Resets all internal counters at next synchro. This should be done only once. It ensures that the mechanism movement will be synchronized with the synchro signal. | 0 | Timing registers free | Timing registers reset at SYNC time |
| 5 | RESET_PLL_BY_BOLC<br>When enabled, the PLL locks on BOLC sync. Otherwise, the PLL is in free run. | 0 | BOLC not connected to PLL. PLL is free running | PLL is locked on BOLC. (for PHOTO mode) |
| 6-16 | Not used | 0 | NA | |
| 17 | SAMPLE and IRQ frequency selection | 00 | Sample frequency in spectro mode | |

| | | | |
|---|---|---|---|
| 18 | | | 00 = 8192 Hz<br>01 = 4096 Hz<br>10 = 2048 Hz<br>11 = 1024 Hz<br>Note: 01 means bit18=0 and bit17=1 |
| 19 | CRE clock frequency | 00 | CRE CLOCK frequency<br>00 = 8192 Hz<br>01 = 4096 Hz<br>10 = 2048 Hz<br>11 = 1024 Hz<br>Note: 10 means bit18=1 and bit17=0 |
| 20 | | | |
| 21-31 | Not used | 0 | NA |

## 4.3 Dump/Check commands

The format of dump and check commands is defined in [AD6].

The activity IDs for these commands are :

Dump : 200

Check : 210

### 4.3.1 Dump addresses

In order to be able to issue a dump command, one should know the start address and length of the memory area to dump.

These addresses are subject to change for every new version of the software and every patch. These addresses will be listed in this document but it is also possible to obtain them directly from the software. By dumping DM at address 0x60000, you will get the first column of the table below.

These addresses can also be used to program the custom hk entries: In exemple, if you want to monitor the DMC_CHOP_IA continuously, you should:

- Get the address of this variable from the table below: 0x346ab
- Get the length in words from the table below: 1
- Get the length in bytes from the HK table: 2
- Configure the custom hk entry 1 to monitor CHOP_IA by writing 5 words in DMC_WRT_DIAG_HK_CONF_TAB:
    - Address = 0x546ad
    - Size in words = 1
    - Size in bytes = 2
    - Function = 0

- Validity = 1

Note: the table below is given for information only and is applicable for version 6.026 only, always use the dump command each time you change the software

| Address in DM | Content | Length (in words) |
|---|---|---|
| 0x00054490 | DMC_WRT_TIME | 2 |
| 0x00054494 | DMC_WRT_SEQ_BUFFER | max 512 |
| 0x00054494 | DMC_WRT_SEQ_BUFFER_0 | max 52 |
| 0x000544c8 | DMC_WRT_SEQ_BUFFER_1 | max 52 |
| 0x000544fc | DMC_WRT_SEQ_BUFFER_2 | max 52 |
| 0x00054530 | DMC_WRT_SEQ_BUFFER_3 | max 52 |
| 0x00054564 | DMC_WRT_SEQ_BUFFER_4 | max 52 |
| 0x00054598 | DMC_WRT_SEQ_BUFFER_5 | max 52 |
| 0x000545cc | DMC_WRT_SEQ_BUFFER_6 | max 52 |
| 0x00054600 | DMC_WRT_SEQ_BUFFER_7 | max 52 |
| 0x00054634 | DMC_WRT_SEQ_BUFFER_8 | max 52 |
| 0x00054668 | DMC_WRT_SEQ_BUFFER_9 | max 44 |
| 0x000036a3 | DMC_WRT_GRAT_MAX_POWER | 1 |
| 0x00054479 | DMC_WRT_DIAG_HK_LIST | max 16 |
| 0x00002d2e | DMC_WRT_DIAG_HK_CONF_TAB | max 50 |
| 0x0000436b | DMC_WRT_GRAT_CONF_PAR | 9 |
| 0x0000439b | DMC_WRT_CHOP_CONF_PAR | 21 |
| 0x000543c3 | DMC_WRT_FW_SPEC_CONF_PAR | 6 |
| 0x000543d7 | DMC_WRT_FW_PHOT_CONF_PAR | 6 |
| 0x000543fe | DMC_WRT_CS1_CONF_PAR | 7 |
| 0x0005443a | DMC_WRT_CS2_CONF_PAR | 7 |
| 0x00004386 | DMC_WRT_NOT_USED_2 | 1 |
| 0x00003d74 | DMC_WRT_BOL_REC_OPT | 1 |
| 0x00003aee | DMC_WRT_B_DEC_REC_OPT | 1 |
| 0x00003c2d | DMC_WRT_R_DEC_REC_OPT | 1 |
| 0x0005513f | DMC_WRT_MAX_DITHER | 1 |
| 0x00003d61 | DMC_WRT_R_SPEC_PAR | 6 |
| 0x00003c22 | DMC_WRT_B_SPEC_PAR | 6 |
| 0x0005475f | DMC_WRT_SPU_TRAN_MODE | 2 |
| 0x0005471b | DMC_WRT_TIMING_FPGA_PAR | max 6 |
| 0x00054761 | DMC_WRT_B_PACKT_ENC_LINK | 1 |
| 0x00054762 | DMC_WRT_R_PACKT_ENC_LINK | 1 |
| 0x00004376 | DMC_WRT_GRAT_INDUCT_AMPL | 1 |
| 0x00004375 | DMC_WRT_GRAT_RANGE | 1 |
| 0x00004377 | DMC_WRT_GRAT_HALL_OFFSET | 1 |
| 0x00004378 | DMC_WRT_GRAT_DEG_MODE_PARAM | 2 |

| | | |
|---|---|---|
| 0x00004388 | DMC_WRT_GRAT_CONF_FILT | 5 |
| 0x00054763 | DMC_DMP_MEM_SCR_STAT<br>Memory Scrubbing Status:<br>- Last DM address checked<br>- Last PM address checked<br>- Index where the next Single Failure in DM will be stored in DMC_DMP_MEM_SCR_SF_DM<br>- Index where the next Single Failure in PM will be stored in DMC_DMP_MEM_SCR_SF_PM<br>- Index where the next Double Failure in DM will be stored in DMC_DMP_MEM_SCR_SF_DM<br>- Index where the next Double Failure in PM will be stored in DMC_DMP_MEM_SCR_SF_PM | 6 |
| 0x00054769 | DMC_DMP_MEM_SCR_SF_DM<br>Array containing the DM addresses where a single failure has been detected. The last failing address is given by the index-1 (the index can be found in DMC_DMP_MEM_SCR_STAT) | 256 |
| 0x00054869 | DMC_DMP_MEM_SCR_SF_PM<br>Array containing the PM addresses where a single failure has been detected. The last failing address is given by the index-1 (the index can be found in DMC_DMP_MEM_SCR_STAT) | 256 |
| 0x00054969 | DMC_DMP_MEM_SCR_DF_DM<br>Array containing the DM addresses where a double failure has been detected. The last failing address is given by the index-1 (the index can be found in DMC_DMP_MEM_SCR_STAT) | 256 |
| 0x00054a69 | DMC_DMP_MEM_SCR_DF_PM<br>Array containing the PM addresses where a double failure has been detected. The last failing address is given by the index-1 (the index can be found in DMC_DMP_MEM_SCR_STAT) | 256 |
| 0x0000427a | BOLC_HK_1 | 1 |
| 0x0000427b | BOLC_HK_2 | 1 |
| 0x0000427c | BOLC_HK_3 | 1 |
| 0x0000427d | BOLC_HK_4 | 1 |
| 0x0000427e | BOLC_HK_5 | 1 |
| 0x0000427f | BOLC_HK_6 | 1 |
| 0x00004280 | BOLC_HK_7 | 1 |
| 0x00004281 | BOLC_HK_8 | 1 |
| 0x00004282 | BOLC_HK_9 | 1 |
| 0x00004283 | BOLC_HK_10 | 1 |
| 0x00004284 | BOLC_HK_11 | 1 |

| | | |
|---|---|---|
| 0x00004285 | BOLC_HK_12 | 1 |
| 0x00004286 | BOLC_HK_13 | 1 |
| 0x00004287 | BOLC_HK_14 | 1 |
| 0x00004288 | BOLC_HK_15 | 1 |
| 0x00004289 | BOLC_HK_16 | 1 |
| 0x0000428a | BOLC_HK_17 | 1 |
| 0x0000428b | BOLC_HK_18 | 1 |
| 0x0000428c | BOLC_HK_19 | 1 |
| 0x0000428d | BOLC_HK_20 | 1 |
| 0x0000428e | BOLC_HK_21 | 1 |
| 0x0000428f | BOLC_HK_22 | 1 |
| 0x00004290 | BOLC_HK_23 | 1 |
| 0x00004291 | BOLC_HK_24 | 1 |
| 0x00004292 | BOLC_HK_25 | 1 |
| 0x00004293 | BOLC_HK_26 | 1 |
| 0x00004294 | BOLC_HK_27 | 1 |
| 0x00004295 | BOLC_HK_28 | 1 |
| 0x00004296 | BOLC_HK_29 | 1 |
| 0x00004297 | BOLC_HK_30 | 1 |
| 0x00004298 | BOLC_HK_31 | 1 |
| 0x00004299 | BOLC_HK_32 | 1 |
| 0x0000429a | BOLC_HK_33 | 1 |
| 0x0000429b | BOLC_HK_34 | 1 |
| 0x0000429c | BOLC_HK_35 | 1 |
| 0x0000429d | BOLC_HK_36 | 1 |
| 0x0000429e | BOLC_HK_37 | 1 |
| 0x0000429f | BOLC_HK_38 | 1 |
| 0x000042a0 | BOLC_HK_39 | 1 |
| 0x000042a1 | BOLC_HK_40 | 1 |
| 0x000042a2 | BOLC_HK_41 | 1 |
| 0x000042a3 | BOLC_HK_42 | 1 |
| 0x000042a4 | BOLC_HK_43 | 1 |
| 0x000042a5 | BOLC_HK_44 | 1 |
| 0x000042a6 | BOLC_HK_45 | 1 |
| 0x000042a7 | BOLC_HK_46 | 1 |
| 0x000042a8 | BOLC_HK_47 | 1 |
| 0x000042a9 | BOLC_HK_48 | 1 |
| 0x000042aa | BOLC_HK_49 | 1 |
| 0x000042ab | BOLC_HK_50 | 1 |
| 0x000042ac | BOLC_HK_51 | 1 |
| 0x000042ad | BOLC_HK_52 | 1 |
| 0x000042ae | BOLC_HK_53 | 1 |
| 0x000042af | BOLC_HK_54 | 1 |
| 0x000042b0 | BOLC_HK_55 | 1 |

| | | |
|---|---|---|
| 0x000042b1 | BOLC_HK_56 | 1 |
| 0x000042b2 | BOLC_HK_57 | 1 |
| 0x000042b3 | BOLC_HK_58 | 1 |
| 0x000042b4 | BOLC_HK_59 | 1 |
| 0x000042b5 | BOLC_HK_60 | 1 |
| 0x000042b6 | BOLC_HK_61 | 1 |
| 0x000042b7 | BOLC_HK_62 | 1 |
| 0x000042b8 | BOLC_HK_63 | 1 |
| 0x000042b9 | BOLC_HK_64 | 1 |
| 0x000042ba | BOLC_HK_65 | 1 |
| 0x000042bb | BOLC_HK_66 | 1 |
| 0x000042bc | BOLC_HK_67 | 1 |
| 0x000042bd | BOLC_HK_68 | 1 |
| 0x000042be | BOLC_HK_69 | 1 |
| 0x000042bf | BOLC_HK_70 | 1 |
| 0x000042c0 | BOLC_HK_71 | 1 |
| 0x000042c1 | BOLC_HK_72 | 1 |
| 0x000042c2 | BOLC_HK_73 | 1 |
| 0x000042c3 | BOLC_HK_74 | 1 |
| 0x000042c4 | BOLC_HK_75 | 1 |
| 0x000042c5 | BOLC_HK_76 | 1 |
| 0x000042c6 | BOLC_HK_77 | 1 |
| 0x000042c7 | BOLC_HK_78 | 1 |
| 0x000042c8 | BOLC_HK_79 | 1 |
| 0x000042c9 | BOLC_HK_80 | 1 |
| 0x000042ca | BOLC_HK_81 | 1 |
| 0x000042cb | BOLC_HK_82 | 1 |
| 0x000042cc | BOLC_HK_83 | 1 |
| 0x000042cd | BOLC_HK_84 | 1 |
| 0x000042ce | BOLC_HK_85 | 1 |
| 0x000042cf | BOLC_HK_86 | 1 |
| 0x000042d0 | BOLC_HK_87 | 1 |
| 0x000042d1 | BOLC_HK_88 | 1 |
| 0x000042d2 | BOLC_HK_89 | 1 |
| 0x000042d3 | BOLC_HK_90 | 1 |
| 0x000042d4 | BOLC_HK_91 | 1 |
| 0x000042d5 | BOLC_HK_92 | 1 |
| 0x000042d6 | BOLC_HK_93 | 1 |
| 0x000042d7 | BOLC_HK_94 | 1 |
| 0x000042d8 | BOLC_HK_95 | 1 |
| 0x000042d9 | BOLC_HK_96 | 1 |
| 0x000042da | BOLC_HK_97 | 1 |
| 0x000042db | BOLC_HK_98 | 1 |
| 0x000042dc | BOLC_HK_99 | 1 |

| | | |
|---|---|---|
| 0x000042dd | BOLC_HK_100 | 1 |
| 0x000042de | BOLC_HK_101 | 1 |
| 0x000042df | BOLC_HK_102 | 1 |
| 0x000042e0 | BOLC_HK_103 | 1 |
| 0x000042e1 | BOLC_HK_104 | 1 |
| 0x000042e2 | BOLC_HK_105 | 1 |
| 0x000042e3 | BOLC_HK_106 | 1 |
| 0x000042e4 | BOLC_HK_107 | 1 |
| 0x000042e5 | BOLC_HK_108 | 1 |
| 0x000042e6 | BOLC_HK_109 | 1 |
| 0x000042e7 | BOLC_HK_110 | 1 |
| 0x000042e8 | BOLC_HK_111 | 1 |
| 0x000042e9 | BOLC_HK_112 | 1 |
| 0x000042ea | BOLC_HK_113 | 1 |
| 0x000042eb | BOLC_HK_114 | 1 |
| 0x000042ec | BOLC_HK_115 | 1 |
| 0x000042ed | BOLC_HK_116 | 1 |
| 0x000042ee | BOLC_HK_117 | 1 |
| 0x000042ef | BOLC_HK_118 | 1 |
| 0x000042f0 | BOLC_HK_119 | 1 |
| 0x000042f1 | BOLC_HK_120 | 1 |
| 0x000042f2 | BOLC_HK_121 | 1 |
| 0x000042f3 | BOLC_HK_122 | 1 |
| 0x000042f4 | BOLC_HK_123 | 1 |
| 0x000042f5 | BOLC_HK_124 | 1 |
| 0x000042f6 | BOLC_HK_125 | 1 |
| 0x000042f7 | BOLC_HK_126 | 1 |
| 0x000042f8 | BOLC_HK_127 | 1 |
| 0x000042f9 | BOLC_HK_128 | 1 |
| 0x000042fa | BOLC_HK_129 | 1 |
| 0x000042fb | BOLC_HK_130 | 1 |
| 0x000042fc | BOLC_HK_131 | 1 |
| 0x000042fd | BOLC_HK_132 | 1 |
| 0x000042fe | BOLC_HK_133 | 1 |
| 0x000042ff | BOLC_HK_134 | 1 |
| 0x00004300 | BOLC_HK_135 | 1 |
| 0x00004301 | BOLC_HK_136 | 1 |
| 0x00004302 | BOLC_HK_137 | 1 |
| 0x00004303 | BOLC_HK_138 | 1 |
| 0x00004304 | BOLC_HK_139 | 1 |
| 0x00004305 | BOLC_HK_140 | 1 |
| 0x00004306 | BOLC_HK_141 | 1 |
| 0x00004307 | BOLC_HK_142 | 1 |
| 0x00004308 | BOLC_HK_143 | 1 |

| | | |
|---|---|---|
| 0x00004309 | BOLC_HK_144 | 1 |
| 0x0000430a | BOLC_HK_145 | 1 |
| 0x0000430b | BOLC_HK_146 | 1 |
| 0x0000430c | BOLC_HK_147 | 1 |
| 0x0000430d | BOLC_HK_148 | 1 |
| 0x0000430e | BOLC_HK_149 | 1 |
| 0x0000430f | BOLC_HK_150 | 1 |
| 0x00004310 | BOLC_HK_151 | 1 |
| 0x00004311 | BOLC_HK_152 | 1 |
| 0x00004312 | BOLC_HK_153 | 1 |
| 0x00004313 | BOLC_HK_154 | 1 |
| 0x00004314 | BOLC_HK_155 | 1 |
| 0x00004315 | BOLC_HK_156 | 1 |
| 0x00004316 | BOLC_HK_157 | 1 |
| 0x00004317 | BOLC_HK_158 | 1 |
| 0x00004318 | BOLC_HK_159 | 1 |
| 0x00004319 | BOLC_HK_160 | 1 |
| 0x0000431a | BOLC_HK_161 | 1 |
| 0x0000431b | BOLC_HK_162 | 1 |
| 0x0000431c | BOLC_HK_163 | 1 |
| 0x0000431d | BOLC_HK_164 | 1 |
| 0x0000431e | BOLC_HK_165 | 1 |
| 0x0000431f | BOLC_HK_166 | 1 |
| 0x00004320 | BOLC_HK_167 | 1 |
| 0x00004321 | BOLC_HK_168 | 1 |
| 0x00004322 | BOLC_HK_169 | 1 |
| 0x00004323 | BOLC_HK_170 | 1 |
| 0x00004324 | BOLC_HK_171 | 1 |
| 0x00004325 | BOLC_HK_172 | 1 |
| 0x00004326 | BOLC_HK_173 | 1 |
| 0x00004327 | BOLC_HK_174 | 1 |
| 0x00004328 | BOLC_HK_175 | 1 |
| 0x00004329 | BOLC_HK_176 | 1 |
| 0x0000432a | BOLC_HK_177 | 1 |
| 0x0000432b | BOLC_HK_178 | 1 |
| 0x0000432c | BOLC_HK_179 | 1 |
| 0x0000432d | BOLC_HK_180 | 1 |
| 0x0000432e | BOLC_HK_181 | 1 |
| 0x0000432f | BOLC_HK_182 | 1 |
| 0x00004330 | BOLC_HK_183 | 1 |
| 0x00004331 | BOLC_HK_184 | 1 |
| 0x00004332 | BOLC_HK_185 | 1 |
| 0x00004333 | BOLC_HK_186 | 1 |
| 0x00004334 | BOLC_HK_187 | 1 |

| | | |
|---|---|---|
| 0x00004335 | BOLC_HK_188 | 1 |
| 0x00004336 | BOLC_HK_189 | 1 |
| 0x00004337 | BOLC_HK_190 | 1 |
| 0x00004338 | BOLC_HK_191 | 1 |
| 0x00004339 | BOLC_HK_192 | 1 |
| 0x0000433a | BOLC_HK_193 | 1 |
| 0x0000433b | BOLC_HK_194 | 1 |
| 0x0000433c | BOLC_HK_195 | 1 |
| 0x0000433d | BOLC_HK_196 | 1 |
| 0x0005475e | DMC_SW_GLOBAL_ST | 1 |
| 0x0005448a | DMC_SEQ_STATUS | 1 |
| 0x00003d6b | DMC_DPU_REC_STAT | 1 |
| 0x00003d6d | DMC_DPU_SEN_STAT | 1 |
| 0x00003aed | DMC_DECB_REC_STA | 1 |
| 0x00003c16 | DMC_DECB_CTRL_ST | 1 |
| 0x0000435a | DMC_BLUE_PAC_ENC | 1 |
| 0x00003c2c | DMC_DECR_REC_STA | 1 |
| 0x00003d55 | DMC_DECR_CTRL_ST | 1 |
| 0x0000435d | DMC_RED_PAC_ENC | 1 |
| 0x00003d73 | DMC_BOL_REC_STAT | 1 |
| 0x00003d6f | DMC_BOL_CTRL_STA | 1 |
| 0x0000436a | DMC_GRAT_CTRL_ST | 1 |
| 0x0000439a | DMC_CHOP_CTRL_ST | 1 |
| 0x000543c2 | DMC_FW_SPEC_CTRL | 1 |
| 0x000543d6 | DMC_FW_PHOT_CTRL | 1 |
| 0x000036a3 | DMC_CHECKSUM | 1 |
| 0x000543ea | DMC_CS1_CTRL_STA | 1 |
| 0x00054426 | DMC_CS2_CTRL_STA | 1 |
| 0x0005448b | DMC_SEQ_OPTIONS | 1 |
| 0x00054493 | DMC_SEQ_POINTER | 1 |
| 0x00054695 | DMC_SEQ_LOOP_ID0 | 1 |
| 0x00054696 | DMC_SEQ_LOOP_ID1 | 1 |
| 0x00054697 | DMC_SEQ_LOOP_ID2 | 1 |
| 0x00054698 | DMC_SEQ_LOOP_ID3 | 1 |
| 0x00054699 | DMC_SEQ_LOOP_ID4 | 1 |
| 0x0005469a | DMC_SEQ_WAIT_IND | 1 |
| 0x00054492 | DMC_SEQ_LABEL | 1 |
| 0x0005448c | DMC_OBSID | 1 |
| 0x0005448d | DMC_BBID | 1 |
| 0x0005448e | DMC_TIME_1 | 1 |
| 0x0005448f | DMC_TIME_2 | 1 |
| 0x00003aef | DMC_DECB_REC_PAC | 1 |
| 0x00003c2e | DMC_DECR_REC_PAC | 1 |
| 0x00003c17 | DMC_DECB_CTRL_PA | 1 |

| | | |
|---|---|---|
| 0x00003d56 | DMC_DECR_CTRL_PA | 1 |
| 0x0000435c | DMC_BLUE_ENC_PAC | 1 |
| 0x0000435f | DMC_RED_ENC_PAC | 1 |
| 0x00003d75 | DMC_BOL_REC_PAC | 1 |
| 0x00003d70 | DMC_BOL_CTRL_PAC | 1 |
| 0x00003d6c | DMC_DPU_REC_PAC | 1 |
| 0x00003d6e | DMC_DPU_SEND_PAC | 1 |
| 0x00003af0 | DMC_B_SPEC_READ | 1 |
| 0x00003c2f | DMC_R_SPEC_READ | 1 |
| 0x0005469e | DMC_BOL_READ_CNT | 1 |
| 0x0005475d | DMC_CPU_LOAD | 1 |
| 0x0005469c | DMC_IRS_CNT | 1 |
| 0x00003aec | DMC_VID | 1 |
| 0x00004392 | DMC_CHOP_CUR_POS | 1 |
| 0x00004396 | DMC_CHOP_SETPOIN | 1 |
| 0x00004399 | DMC_CHOP_TARGET | 1 |
| 0x00004394 | DMC_CHOP_PID_ERR | 1 |
| 0x00004395 | DMC_CHOP_PID_ACC | 1 |
| 0x0005513f | DMC_CHOP_MAX_DIT | 1 |
| 0x00004362 | DMC_GRAT_CUR_POS | 1 |
| 0x00004366 | DMC_GRAT_SETPOIN | 1 |
| 0x00004369 | DMC_GRAT_TARGET | 1 |
| 0x00004364 | DMC_GRAT_PID_ERR | 1 |
| 0x00004365 | DMC_GRAT_PID_ACC | 1 |
| 0x000543d3 | DMC_FWSP_CUR_POS | 1 |
| 0x00003acd | DMC_FWGRAT_HALLA | 1 |
| 0x00003ace | DMC_FWGRAT_HALLB | 1 |
| 0x00004393 | DMC_CHOP_OUTPUT | 1 |
| 0x0005469d | DMC_ISR_STAT | 1 |
| 0x000543e7 | DMC_FWPH_CUR_POS | 1 |
| 0x000036a3 | DMC_SPARE1 | 1 |
| 0x000036a3 | DMC_SPARE2 | 1 |
| 0x00054b6a | DMC_PLL_RES_LO | 1 |
| 0x00054b6b | DMC_PLL_RES_HI | 1 |
| 0x00003b26 | DMC_DECB_VDDD_3 | 1 |
| 0x00003b2c | DMC_DECB_VSS_3 | 1 |
| 0x00003b2d | DMC_DECB_VGND_3 | 1 |
| 0x00003b2a | DMC_DECB_VCAN1_3 | 1 |
| 0x00003b29 | DMC_DECB_VCAN2_3 | 1 |
| 0x00003b2e | DMC_DECB_V0BIAS3 | 1 |
| 0x00003b2f | DMC_DECB_VBI_R_3 | 1 |
| 0x00003b24 | DMC_DECB_V0V_3 | 1 |
| 0x00003b28 | DMC_DECB_VSCP_3 | 1 |
| 0x00003b2b | DMC_DECB_VDDR_3 | 1 |

| | | |
|---|---|---|
| 0x00003b27 | DMC_DECB_VDDA_3 | 1 |
| 0x00003b25 | DMC_DECB_VWELL_3 | 1 |
| 0x00003b31 | DMC_DECB_IDDA_3 | 1 |
| 0x00003b30 | DMC_DECB_IDDD_3 | 1 |
| 0x00003b32 | DMC_DECB_ISS_3 | 1 |
| 0x00003b33 | DMC_DECB_IGND_3 | 1 |
| 0x00003b15 | DMC_DECB_HEAT_C | 1 |
| 0x00003b16 | DMC_DECB_HEAT_V | 1 |
| 0x00003b14 | DMC_DECB_REF_0V3 | 1 |
| 0x00003b1d | DMC_DECB_DCDC_T3 | 1 |
| 0x000036a3 | DMC_DECB_SPARE5 | 1 |
| 0x00003b20 | DMC_DECB_DCDC_P5 | 1 |
| 0x00003b21 | DMC_DECB_AC_CUR | 1 |
| 0x00003b4a | DMC_DECB_TS_ST_3 | 1 |
| 0x00003af2 | DMC_DECB_CL_RO_3 | 1 |
| 0x00003af4 | DMC_DECB_RO_RA_3 | 1 |
| 0x00003af6 | DMC_DECB_CR_ST_3 | 1 |
| 0x00003af8 | DMC_DECB_BR_CM_3 | 1 |
| 0x00003afa | DMC_DECB_ZB_CM_3 | 1 |
| 0x00003afc | DMC_DECB_SR_RB_3 | 1 |
| 0x00003b48 | DMC_DECB_TS_1_3 | 1 |
| 0x00003b49 | DMC_DECB_TS_2_3 | 1 |
| 0x00003b02 | DMC_DECB_RO_CO_3 | 1 |
| 0x00003b82 | DMC_DECB_RA_CO_3 | 1 |
| 0x00003b5d | DMC_DECB_VDDD_4 | 1 |
| 0x00003b63 | DMC_DECB_VSS_4 | 1 |
| 0x00003b64 | DMC_DECB_VGND_4 | 1 |
| 0x00003b61 | DMC_DECB_VCAN1_4 | 1 |
| 0x00003b60 | DMC_DECB_VCAN2_4 | 1 |
| 0x00003b65 | DMC_DECB_V0BIAS4 | 1 |
| 0x00003b66 | DMC_DECB_VBI_R_4 | 1 |
| 0x00003b5b | DMC_DECB_V0V_4 | 1 |
| 0x00003b5f | DMC_DECB_VSCP_4 | 1 |
| 0x00003b62 | DMC_DECB_VDDR_4 | 1 |
| 0x00003b5e | DMC_DECB_VDDA_4 | 1 |
| 0x00003b5c | DMC_DECB_VWELL_4 | 1 |
| 0x00003b68 | DMC_DECB_IDDA_4 | 1 |
| 0x00003b67 | DMC_DECB_IDDD_4 | 1 |
| 0x00003b69 | DMC_DECB_ISS_4 | 1 |
| 0x00003b6a | DMC_DECB_IGND_4 | 1 |
| 0x00003b4c | DMC_DECB_FLASH_C | 1 |
| 0x00003b4d | DMC_DECB_FLASH_V | 1 |
| 0x00003b4b | DMC_DECB_REF_0V4 | 1 |
| 0x00003b54 | DMC_DECB_DCDC_T4 | 1 |

| | | |
|---|---|---|
| 0x000036a3 | DMC_DECB_SPARE5B | 1 |
| 0x00003b57 | DMC_DECB_DCDC_P15 | 1 |
| 0x00003b58 | DMC_DECB_DCDC_N15 | 1 |
| 0x00003b81 | DMC_DECB_TS_ST_4 | 1 |
| 0x00003af3 | DMC_DECB_CL_RO_4 | 1 |
| 0x00003af5 | DMC_DECB_RO_RA_4 | 1 |
| 0x00003af7 | DMC_DECB_CR_ST_4 | 1 |
| 0x00003af9 | DMC_DECB_BR_CM_4 | 1 |
| 0x00003afb | DMC_DECB_ZB_CM_4 | 1 |
| 0x00003afd | DMC_DECB_SR_RB_4 | 1 |
| 0x00003b7f | DMC_DECB_TS_1_4 | 1 |
| 0x00003b80 | DMC_DECB_TS_2_4 | 1 |
| 0x00003b03 | DMC_DECB_RO_CO_4 | 1 |
| 0x00003b83 | DMC_DECB_RA_CO_4 | 1 |
| 0x00003c65 | DMC_DECR_VDDD_1 | 1 |
| 0x00003c6b | DMC_DECR_VSS_1 | 1 |
| 0x00003c6c | DMC_DECR_VGND_1 | 1 |
| 0x00003c69 | DMC_DECR_VCAN1_1 | 1 |
| 0x00003c68 | DMC_DECR_VCAN2_1 | 1 |
| 0x00003c6d | DMC_DECR_V0BIAS1 | 1 |
| 0x00003c6e | DMC_DECR_VBI_R_1 | 1 |
| 0x00003c63 | DMC_DECR_V0V_1 | 1 |
| 0x00003c67 | DMC_DECR_VSCP_1 | 1 |
| 0x00003c6a | DMC_DECR_VDDR_1 | 1 |
| 0x00003c66 | DMC_DECR_VDDA_1 | 1 |
| 0x00003c64 | DMC_DECR_VWELL_1 | 1 |
| 0x00003c70 | DMC_DECR_IDDA_1 | 1 |
| 0x00003c6f | DMC_DECR_IDDD_1 | 1 |
| 0x00003c71 | DMC_DECR_ISS_1 | 1 |
| 0x00003c72 | DMC_DECR_IGND_1 | 1 |
| 0x00003c54 | DMC_DECR_HEAT_C | 1 |
| 0x00003c55 | DMC_DECR_HEAT_V | 1 |
| 0x00003c53 | DMC_DECR_REF_0V_1 | 1 |
| 0x00003c5c | DMC_DECR_DCDC_T1 | 1 |
| 0x000036a3 | DMC_DECR_SPARE5 | 1 |
| 0x00003c5f | DMC_DECR_DCDC_P5 | 1 |
| 0x00003c60 | DMC_DECR_AC_CUR | 1 |
| 0x00003c89 | DMC_DECR_TS_ST_1 | 1 |
| 0x00003c31 | DMC_DECR_CL_RO_1 | 1 |
| 0x00003c33 | DMC_DECR_RO_RA_1 | 1 |
| 0x00003c35 | DMC_DECR_CR_ST_1 | 1 |
| 0x00003c37 | DMC_DECR_BR_CM_1 | 1 |
| 0x00003c39 | DMC_DECR_ZB_CM_1 | 1 |
| 0x00003c3b | DMC_DECR_SR_RB_1 | 1 |

| | | |
|---|---|---|
| 0x00003c87 | DMC_DECR_TS_1_1 | 1 |
| 0x00003c88 | DMC_DECR_TS_2_1 | 1 |
| 0x00003c41 | DMC_DECR_RO_CO_1 | 1 |
| 0x00003cc1 | DMC_DECR_RA_CO_1 | 1 |
| 0x00003c9c | DMC_DECR_VDDD_2 | 1 |
| 0x00003ca2 | DMC_DECR_VSS_2 | 1 |
| 0x00003ca3 | DMC_DECR_VGND_2 | 1 |
| 0x00003ca0 | DMC_DECR_VCAN1_2 | 1 |
| 0x00003c9f | DMC_DECR_VCAN2_2 | 1 |
| 0x00003ca4 | DMC_DECR_V0BIAS2 | 1 |
| 0x00003ca5 | DMC_DECR_VBI_R_2 | 1 |
| 0x00003c9a | DMC_DECR_V0V_2 | 1 |
| 0x00003c9e | DMC_DECR_VSCP_2 | 1 |
| 0x00003ca1 | DMC_DECR_VDDR_2 | 1 |
| 0x00003c9d | DMC_DECR_VDDA_2 | 1 |
| 0x00003c9b | DMC_DECR_VWELL_2 | 1 |
| 0x00003ca7 | DMC_DECR_IDDA_2 | 1 |
| 0x00003ca6 | DMC_DECR_IDDD_2 | 1 |
| 0x00003ca8 | DMC_DECR_ISS_2 | 1 |
| 0x00003ca9 | DMC_DECR_IGND_2 | 1 |
| 0x00003c8b | DMC_DECR_FLASH_C | 1 |
| 0x00003c8c | DMC_DECR_FLASH_V | 1 |
| 0x00003c8a | DMC_DECR_REF_0V2 | 1 |
| 0x00003c93 | DMC_DECR_DCDC_T2 | 1 |
| 0x000036a3 | DMC_DECR_SPARE5B | 1 |
| 0x00003c96 | DMC_DECR_DCDC_P15 | 1 |
| 0x00003c97 | DMC_DECR_DCDC_N15 | 1 |
| 0x00003cc0 | DMC_DECR_TS_ST_2 | 1 |
| 0x00003c32 | DMC_DECR_CL_RO_2 | 1 |
| 0x00003c34 | DMC_DECR_RO_RA_2 | 1 |
| 0x00003c36 | DMC_DECR_CR_ST_2 | 1 |
| 0x00003c38 | DMC_DECR_BR_CM_2 | 1 |
| 0x00003c3a | DMC_DECR_ZB_CM_2 | 1 |
| 0x00003c3c | DMC_DECR_SR_RB_2 | 1 |
| 0x00003cbe | DMC_DECR_TS_1_2 | 1 |
| 0x00003cbf | DMC_DECR_TS_2_2 | 1 |
| 0x00003c42 | DMC_DECR_RO_CO_2 | 1 |
| 0x00003cc2 | DMC_DECR_RA_CO_2 | 1 |
| 0x000036a3 | DMC_SPARE4 | 1 |
| 0x000036a3 | DMC_SPARE5 | 1 |
| 0x000036a3 | DMC_SPARE6 | 1 |
| 0x00054755 | DMC_FPU_T_SENS_ST | 1 |
| 0x0005475b | DMC_FW_SPEC_TEMP | 1 |
| 0x0005475c | DMC_FW_PHOT_TEMP | 1 |

| | | |
|---|---|---|
| 0x00054756 | DMC_CHOPPER_TEMP | 1 |
| 0x0005475a | DMC_GRATING_TEMP | 1 |
| 0x000546a4 | DMC_PSC_V1 | 1 |
| 0x000546ac | DMC_PSC_V2 | 1 |
| 0x000546b4 | DMC_PSC_V3 | 1 |
| 0x000546bc | DMC_PSC_V4 | 1 |
| 0x000546a8 | DMC_DCDC_TEMP | 1 |
| 0x000546a9 | DMC_DSP_TEMP | 1 |
| 0x000036a3 | DMC_SPARE10 | 1 |
| 0x000036a3 | DMC_SPARE11 | 1 |
| 0x000036a3 | DMC_SPARE12 | 1 |
| 0x000036a3 | DMC_SPARE13 | 1 |
| 0x000546b1 | DMC_SPU_PSU_P15V | 1 |
| 0x000546aa | DMC_SPU_SWL_TEMP | 1 |
| 0x000546b2 | DMC_SPU_LWL_TEMP | 1 |
| 0x000546ba | DMC_SPU_PS_TEMP | 1 |
| 0x000546b0 | DMC_SPU_VCC_CUR | 1 |
| 0x000546bf | DMC_SPU_VCC_VOL | 1 |
| 0x000546c1 | DMC_SPU_VP_CUR | 1 |
| 0x00054758 | DMC_FPU_T1_T | 1 |
| 0x00054759 | DMC_FPU_T2_T | 1 |
| 0x000546d0 | DMC_REF_VOLT_0V | 1 |
| 0x00054757 | DMC_CAL_SRC_TEMP | 1 |
| 0x000546c0 | DMC_REF_VOLT_5V | 1 |
| 0x000036a3 | DMC_SPARE16 | 1 |
| 0x000036a3 | DMC_SPARE17 | 1 |
| 0x0005469b | DMC_CUSTOM_ENT_1 | 1 |
| 0x000038a5 | DMC_CUSTOM_ENT_2 | 1 |
| 0x000038a6 | DMC_CUSTOM_ENT_3 | 1 |
| 0x000038a7 | DMC_CUSTOM_ENT_4 | 1 |
| 0x000038a8 | DMC_CUSTOM_ENT_5 | 1 |
| 0x000038a9 | DMC_CUSTOM_ENT_6 | 1 |
| 0x000038aa | DMC_CUSTOM_ENT_7 | 1 |
| 0x000038ab | DMC_CUSTOM_ENT_8 | 1 |
| 0x000038ac | DMC_CUSTOM_ENT_9 | 1 |
| 0x000038ad | DMC_CUSTOM_ENT10 | 1 |
| 0x00004360 | DMC_DET_SIM_STAT | 1 |
| 0x00004361 | DMC_DET_SIM_PER | 1 |
| 0x00054405 | DMC_CS1_RES_VALUE | 1 |
| 0x000543eb | DMC_CS1_OUTPUT | 1 |
| 0x00054441 | DMC_CS2_RES_VALUE | 1 |
| 0x00054427 | DMC_CS2_OUTPUT | 1 |
| 0x00004279 | DMC_BOLC_STATUS | 1 |
| 0x0005475f | DMC_BSPU_TR_MODE | 1 |

| | | |
|---|---|---|
| 0x00054760 | DMC_RSPU_TR_MODE | 1 |
| 0x00004363 | DMC_GRAT_OUTPUT | 1 |
| 0x0005469f | DMC_OBT_COUNT | 1 |
| 0x000546a1 | DMC_MIM_ST | 1 |
| 0x00054765 | DMC_DM_SF_IND | 1 |
| 0x00054766 | DMC_PM_SF_IND | 1 |
| 0x00054767 | DMC_DM_DF_IND | 1 |
| 0x00054768 | DMC_PM_DF_IND | 1 |
| 0x000543ee | DMC_CS1_TARGET | 1 |
| 0x0005442a | DMC_CS2_TARGET | 1 |
| 0x00054475 | DMC_HK_CTRL_STAT | 1 |
| 0x00054476 | DMC_HK_DIAG_STAT | 1 |
| 0x00054477 | DMC_HK_DIAG_PERI | 1 |
| 0x000024b7 | DMC_LAST_ERR_ID | 1 |
| 0x000024a7 | DMC_LAST_ER_BF1 | 1 |
| 0x000024a8 | DMC_LAST_ER_BF2 | 1 |
| 0x000024a9 | DMC_LAST_ER_BF3 | 1 |
| 0x000024aa | DMC_LAST_ER_BF4 | 1 |
| 0x000024ab | DMC_LAST_ER_BF5 | 1 |
| 0x000024ac | DMC_LAST_ER_BF6 | 1 |
| 0x000024ad | DMC_LAST_ER_BF7 | 1 |
| 0x000024ae | DMC_LAST_ER_BF8 | 1 |
| 0x000024af | DMC_LAST_ER_BF9 | 1 |
| 0x000024b0 | DMC_LAST_ER_BF10 | 1 |
| 0x000024b1 | DMC_LAST_ER_BF11 | 1 |
| 0x000024b2 | DMC_LAST_ER_BF12 | 1 |
| 0x000024b3 | DMC_LAST_ER_BF13 | 1 |
| 0x000024b4 | DMC_LAST_ER_BF14 | 1 |
| 0x000024b5 | DMC_LAST_ER_BF15 | 1 |
| 0x000024b6 | DMC_LAST_ER_BF16 | 1 |
| 0x0000433e | BOLC_HK_197 | 1 |
| 0x0000433f | BOLC_HK_198 | 1 |
| 0x00004340 | BOLC_HK_199 | 1 |
| 0x00004341 | BOLC_HK_200 | 1 |
| 0x00004342 | BOLC_HK_201 | 1 |
| 0x00004343 | BOLC_HK_202 | 1 |
| 0x00004344 | BOLC_HK_203 | 1 |
| 0x00004345 | BOLC_HK_204 | 1 |
| 0x00004346 | BOLC_HK_205 | 1 |
| 0x00004347 | BOLC_HK_206 | 1 |
| 0x00004348 | BOLC_HK_207 | 1 |
| 0x00004349 | BOLC_HK_208 | 1 |
| 0x0000434a | BOLC_HK_209 | 1 |
| 0x0000434b | BOLC_HK_210 | 1 |

| | | |
|---|---|---|
| 0x0000434c | BOLC_HK_211 | 1 |
| 0x0000434d | BOLC_HK_212 | 1 |
| 0x0000434e | BOLC_HK_213 | 1 |
| 0x0000434f | BOLC_HK_214 | 1 |
| 0x00004350 | BOLC_HK_215 | 1 |
| 0x00004351 | BOLC_HK_216 | 1 |
| 0x00004352 | BOLC_HK_217 | 1 |
| 0x00004353 | BOLC_HK_218 | 1 |
| 0x00004354 | BOLC_HK_219 | 1 |
| 0x00004355 | BOLC_HK_220 | 1 |
| 0x00004356 | BOLC_HK_221 | 1 |
| 0x00004357 | BOLC_HK_222 | 1 |
| 0x00004358 | BOLC_HK_223 | 1 |
| 0x00004359 | BOLC_HK_224 | 1 |
| 0x000036a3 | LAST_NOMINAL_HK_VALUE | 1 |
| 0x000036a3 | SPARE1_NOMINAL_HK_VALUE | 1 |
| 0x000036a3 | SPARE2_NOMINAL_HK_VALUE | 1 |
| 0x00054708 | DMC_GR_IND_READ | 1 |
| 0x00054703 | DMC_GR_TURN_CAR | 1 |
| 0x00054707 | DMC_GR_PER_CAR | 1 |
| 0x00004382 | DMC_GR_DEG_POS | 1 |
| 0x000036a3 | DMC_SPARE_DIAG7 | 1 |
| 0x000036a3 | DMC_SPARE_DIAG8 | 1 |
| 0x000036a3 | DMC_SPARE_DIAG1 | 1 |
| 0x000036a3 | DMC_SPARE_DIAG2 | 1 |
| 0x000036a3 | DMC_SPARE_DIAG3 | 1 |
| 0x000036a3 | DMC_SPARE_DIAG4 | 1 |
| 0x0005440e | DMC_CS1_VOLT_0V | 1 |
| 0x0005440f | DMC_CS1_VOLT_N5V | 1 |
| 0x00054410 | DMC_CS1_VOLT_P5V | 1 |
| 0x00054411 | DMC_CS1_VOLT_DAC_OUT | 1 |
| 0x00054412 | DMC_CS1_VOLT_SG | 1 |
| 0x00054413 | DMC_CS1_VOLT_BG | 1 |
| 0x00054414 | DMC_CS1_CUR_SG | 1 |
| 0x00054415 | DMC_CS1_CUR_BG | 1 |
| 0x000036a3 | DMC_CS1_SPARE1 | 1 |
| 0x000036a3 | DMC_CS1_SPARE2 | 1 |
| 0x000036a3 | DMC_CS1_SPARE3 | 1 |
| 0x000036a3 | DMC_CS1_SPARE4 | 1 |
| 0x000036a3 | DMC_CS1_SPARE5 | 1 |
| 0x000036a3 | DMC_CS1_SPARE6 | 1 |
| 0x000036a3 | DMC_CS1_SPARE7 | 1 |
| 0x000036a3 | DMC_CS1_SPARE8 | 1 |
| 0x0005444a | DMC_CS2_VOLT_0V | 1 |

| | | |
|---|---|---|
| 0x0005444b | DMC_GR_IND_SINE | 1 |
| 0x0005444c | DMC_GR_IND_COS | 1 |
| 0x0005444d | DMC_CS2_VOLT_DAC_OUT | 1 |
| 0x0005444e | DMC_CS2_VOLT_SG | 1 |
| 0x0005444f | DMC_CS2_VOLT_BG | 1 |
| 0x00054450 | DMC_CS2_CUR_SG | 1 |
| 0x00054451 | DMC_CS2_CUR_BG | 1 |
| 0x000546be | DMC_GR_LL1_CUR | 1 |
| 0x000036a3 | DMC_CS2_SPARE2 | 1 |
| 0x000036a3 | DMC_CS2_SPARE3 | 1 |
| 0x000036a3 | DMC_CS2_SPARE4 | 1 |
| 0x000036a3 | DMC_CS2_SPARE5 | 1 |
| 0x000036a3 | DMC_CS2_SPARE6 | 1 |
| 0x000036a3 | DMC_CS2_SPARE7 | 1 |
| 0x000036a3 | DMC_CS2_SPARE8 | 1 |
| 0x000546a3 | DMC_PSU_5V_VOLT | 1 |
| 0x000546a5 | DMC_FWSPEC_POS_A | 1 |
| 0x000546a6 | DMC_FW_GR_VMOTA | 1 |
| 0x000546a7 | DMC_CHOP_VA | 1 |
| 0x000546ab | DMC_PSU_P15V_V | 1 |
| 0x000546ad | DMC_FWSPEC_POS_B | 1 |
| 0x000546ae | DMC_FW_GR_IMOTA | 1 |
| 0x000546af | DMC_CHOP_IA | 1 |
| 0x000546b3 | DMC_PSU_N15V_V | 1 |
| 0x000546b5 | DMC_FWPHOT_POS_A | 1 |
| 0x000546b6 | DMC_FW_GR_VMOTB | 1 |
| 0x000546b7 | DMC_CHOP_VB | 1 |
| 0x000546b8 | DMC_ADC_VOLT | 1 |
| 0x000546b9 | DMC_FW_GR_IMOTB | 1 |
| 0x000546bb | DMC_PSU_P28V_V | 1 |
| 0x000546bd | DMC_FWPHOT_POS_B | 1 |
| 0x000546c2 | DMC_GR_LL2_CUR | 1 |
| 0x000546c4 | DMC_T_SE_SRC1_LG | 1 |
| 0x000546c3 | DMC_T_SE_SRC1_HG | 1 |
| 0x000546d1 | DMC_T_SE_SRC1_V1 | 1 |
| 0x000546d2 | DMC_T_SE_SRC1_V2 | 1 |
| 0x000546cc | DMC_T_SE_SRC2_LG | 1 |
| 0x000546cb | DMC_T_SE_SRC2_HG | 1 |
| 0x000546c9 | DMC_T_SE_SRC2_V1 | 1 |
| 0x000546ca | DMC_T_SE_SRC2_V2 | 1 |
| 0x00003b1b | DMC_DB_TS12CBS_3 | 1 |
| 0x00003b1c | DMC_DB_TS12CSS_3 | 1 |
| 0x00003b17 | DMC_DECB_TS1_V_3 | 1 |
| 0x00003b19 | DMC_DECB_TS2_V_3 | 1 |

| | | |
|---|---|---|
| 0x00003b18 | DMC_DECB_PS_GEN3 | 1 |
| 0x00003b1a | DMC_DECB_NS_GEN3 | 1 |
| 0x00003b1e | DMC_DECB_D5V_3 | 1 |
| 0x00003b1f | DMC_DECB_D2_5V_3 | 1 |
| 0x00003b22 | DMC_DECB_A5V_3 | 1 |
| 0x00003b23 | DMC_DECB_R5V_3 | 1 |
| 0x00003b52 | DMC_DB_TS12CBS_4 | 1 |
| 0x00003b53 | DMC_DB_TS12CSS_4 | 1 |
| 0x00003b4e | DMC_DECB_TS1_V_4 | 1 |
| 0x00003b50 | DMC_DECB_TS2_V_4 | 1 |
| 0x00003b4f | DMC_DECB_PS_GEN4 | 1 |
| 0x00003b51 | DMC_DECB_NS_GEN4 | 1 |
| 0x00003b55 | DMC_DB_DC_P15V_4 | 1 |
| 0x00003b56 | DMC_DB_DC_N15V_4 | 1 |
| 0x00003b59 | DMC_DECB_A5V_4 | 1 |
| 0x00003b5a | DMC_DECB_R5V_4 | 1 |
| 0x00003c5a | DMC_DR_TS12CBS_1 | 1 |
| 0x00003c5b | DMC_DR_TS12CSS_1 | 1 |
| 0x00003c56 | DMC_DECR_TS1_V_1 | 1 |
| 0x00003c58 | DMC_DECR_TS2_V_1 | 1 |
| 0x00003c57 | DMC_DECR_PS_GEN1 | 1 |
| 0x00003c59 | DMC_DECR_NS_GEN1 | 1 |
| 0x00003c5d | DMC_DECR_D5V_1 | 1 |
| 0x00003c5e | DMC_DECR_D2_5V_1 | 1 |
| 0x00003c61 | DMC_DECR_A5V_1 | 1 |
| 0x00003c62 | DMC_DECR_R5V_1 | 1 |
| 0x00003c91 | DMC_DR_TS12CBS_2 | 1 |
| 0x00003c92 | DMC_DR_TS12CSS_2 | 1 |
| 0x00003c8d | DMC_DECR_TS1_V_2 | 1 |
| 0x00003c8f | DMC_DECR_TS2_V_2 | 1 |
| 0x00003c8e | DMC_DECR_PS_GEN2 | 1 |
| 0x00003c90 | DMC_DECR_NS_GEN2 | 1 |
| 0x00003c94 | DMC_DR_DCDC_P15V_2 | 1 |
| 0x00003c95 | DMC_DR_DCDC_N15V_2 | 1 |
| 0x00003c98 | DMC_DECR_A5V_2 | 1 |
| 0x00003c99 | DMC_DECR_R5V_2 | 1 |
| 0x000546ce | DMC_TS_FW_SPEC_V | 1 |
| 0x000546cf | DMC_TS_FW_PHOT_V | 1 |
| 0x000546cd | DMC_TS_GRAT_V | 1 |
| 0x000546c5 | DMC_TS_CHOP_V | 1 |
| 0x000546c7 | DMC_TS_FPU_T1_V | 1 |
| 0x000546c8 | DMC_TS_FPU_T2_V | 1 |
| 0x000546c6 | DMC_TS_BB_V | 1 |

Reminder note for DMC developpers:

Procedure to obtain this table from SimDPU: It is not possible to dump more than one packet at a time from SimDPU so, this should be done in 3 packets:

Dump from 0x60000, size 240, rename dump.dat into dump1.dat

Dump from 0x600F0, size 240, rename dump.dat into dump2.dat

Dump from 0x601E0, size 240, rename dump.dat into dump3.dat

## 4.4 Using the commands

### 4.4.1 Detector software simulator

#### 4.4.1.1 Simulating readouts

You can ask each of the detector receiver tasks to forward a simulated readout instead of the real readouts provided by the real detectors. This can be done quite easily thanks to the *DMC_START_DET_SIMULATOR* command with the following arguments :

- the 4 MSB identifying the detector to be simulated
- the 28 LSB set to 0.

Each time a readout is received, only the scientific data are replaced by the simulated readout. The header remains the same.

You can simulate the readouts of the Blue and Red DEC in the same time (by setting 3 as the detector identifier of the *DMC_START_DET_SIMULATOR* command).

To stop the simulation, send the *DMC_STOP_DET_SIMULATOR* command (without arguments). Note : this command stops the simulation of all the detectors.

#### 4.4.1.2 Simulating readouts and timing

You can also ask one of the detector receiver tasks to forward a simulated readout with a timing different from the one provided by the real detectors. This is very useful during software development (when real detectors are not available). It may not be useful in flight (TBC).

A task is dedicated to the simulation of one of the detectors (Blue DEC, Red DEC or BOLC). It can simulate any detector. The detector being simulated must be powered-off so it does not send anything on its Spacewire link.

To simulate a detector, the following steps must be followed :

1. Start with DEC/MEC and all detectors powered off.
2. power-on DEC/MEC as usual but don't power-on the detector you want to simulate.
3. Send a trigger command *DMC_START_DET_SIMULATOR* with
   - the 4 MSB indentifying the detector to be simulated
   - the 28 LSB is the period (in ms) between two readouts (period must be greater than 24 when BOLC is simulated and greater than 3 otherwise).

4. Send a trigger command *DMC_STOP_DET_SIMULATOR* when finished.

Then, if you want to use the real detectors, you need to switch-off DEC/MEC and restart it.

### 4.4.2 Detector hardware simulation

2 bits of Cre_ctrl_reg are used for read-out/ramp simulation:

| Bit6: Simulation | Bit9: Ramp simulation | |
|---|---|---|
| 0 | 0 | Nominal behaviour, no simulation |
| 0 | 1 | Nominal behaviour, no simulation |
| 1 | 0 | Every pixel of every readout has the same value and is equal to simul_reg.<br>All analog HK is equal to 0xAA55 or 0x55AA.<br>The digital HK is representative. |
| 1 | 1 | Ramps are simulated. In the first readout of the ramp, all the pixel values are equal to Simul_reg. In the following readouts, this value is decremented by 0x20 at each readout.<br>All analog HK is equal to 0xAA55 or 0x55AA.<br>The digital HK is representative. |

### 4.4.3 Switching between Spectrometry/Photometry modes

At start-up, the DEC/MEC OBS is not configured in any mode. That means that no data from the DECs or BOLC are forwarded to the SPU.

To switch from Spectrometry to Photometry, one should :

1. Stop forwarding the data from Blue DEC to SPU (set bit 2 of *DMC_WRT_B_DEC_REC_OPT* to 1)
2. Stop forwarding the data from Red DEC to SPU (set bit 2 of *DMC_WRT_R_DEC_REC_OPT* to 1)
3. Upload the DMC_WRT_TIMING_FPGA_PAR with: {0, 0, 0, 26, 0x0977602A, 0x32}
4. Apply the new parameters: DMC_SET_TIMING_FPGA_PAR
5. Change the synchronisation signal that is used by the sequencer (*DMC_SYNCHRONIZE_ON_DET* with 4 as argument)
6. Wait 1 second to make sure that DMC has received at least one synchro signal from BOLC and has reset all its internal counters (it will ensure that the mechanisms movement will be synchronized with the synchro signal)
7. Upload the DMC_WRT_TIMING_FPGA_PAR with: {0, 0, 0, 26, 0x0977602A, 0x22}
8. Apply the new parameters: DMC_SET_TIMING_FPGA_PAR

9. Start forwarding the data from BOL to SPU (set bit 2 of *DMC_WRT_BOL_REC_OPT* to 0)

To switch from Photometry to Spectrometry, one should :

1. Stop forwarding the data from BOL to SPU (set bit 2 of *DMC_WRT_B_DEC_REC_OPT* to 1)
2. Upload the DMC_WRT_TIMING_FPGA_PAR with: {0, 0, 0, 26, 0x095217CB, 0xB}
3. Apply the new parameters: DMC_SET_TIMING_FPGA_PAR
4. Change the synchronisation signal that is used by the sequencer (*DMC_SYNCHRONIZE_ON_DET* with 1 or 2 as argument)
5. Start forwarding the data from Blue DEC to SPU (set bit 2 of *DMC_WRT_B_DEC_REC_OPT* to 0)
6. Start forwarding the data from Red DEC to SPU (set bit 2 of *DMC_WRT_R_DEC_REC_OPT* to 0)

Note that, in Spectrometry mode, each detector can also be used separately (in this case, the other one has the Bit 2 of its option field set to 1).

### 4.4.4 Using the spectroscopy detectors

To start using the blue spectroscopy detectors, one should :

1. Switch on the blue DEC (*DMC_SWON_B_DEC*)
2. Switch on the blue spectro array (*DMC_SWON_B_SPEC*)
3. Configure the detector timing (*DMC_WRT_B_SPEC_PAR* and then *DMC_SET_PAR_B_SPEC*)
4. Start forwarding the data to SPU (set bit 2 of *DMC_WRT_B_DEC_REC_OPT* to 0)
5. When done, before switching off the spectro arrays, configure the readouts/ramp to a value smaller than TBD. We recommend to use the default parameters. (*DMC_WRT_B_SPEC_PAR* and then *DMC_SET_PAR_B_SPEC*).
6. switch off the blue spectro array (*DMC_SWOF_B_SPEC*)
7. switch off the blue DEC (*DMC_SWOF_B_DEC*)

Note:

Everytime you switch-on a DEC, a master reset is performed. The master reset interrupts the clock provided to DEC FPGAs and resets them. It means that, if a master-reset is performed while CREs are ON, they are switched-off brutally which is not recommended.

To avoid it, it is forbidden to switch-on a DEC while the other CREs are already ON.

Typical switch-on would then be:
- switch-on blue DEC (includes a master reset)
- switch-on red DEC (includes a master reset)
- switch-on blue CREs
- switch-on red CREs

- send_param_both

If you are using blue DEC only and then, want to switch-on red DEC, you would then need to:
- switch-off blue CREs
- switch-on red DEC
- switch-on blue CREs
- switch-on red CREs
- send_param_both

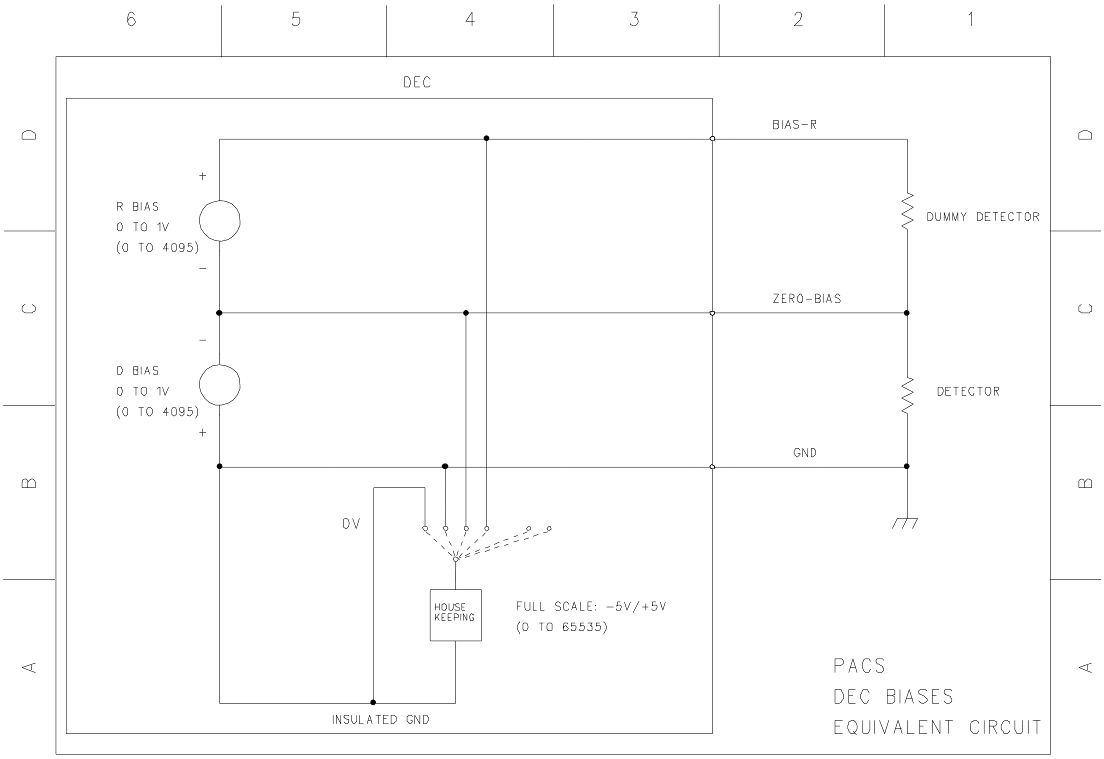
#### 4.4.4.1 Switching-on the CREs

In the text below, we present the procedure to switch-on blue CREs array. The same procedure can easily be adapted to red CREs array.

1. DMC_SWON_B_DEC to switch on the DEC
2. Wait 5 seconds to get the 1355 connection between DEC and CPU board
3. Send the complete set of default parameters. First send a DMC_WRT_B_SPEC_PAR command with these values (in hex: 20-8-18C-0-0-EA60). Then, send a DMC_SET_PAR_B_SPEC to really send the parameters to the DEC.
4. DMC_SWON_B_SPEC to switch on the detector array
5. Wait 15 seconds to let the switch on procedure run and all voltages become stables
6. Then, to activate the CREs (signal SELECT on the CREs), send the first 3 parameters. First send a DMC_WRT_B_SPEC_PAR command with these values (length: 3, values in hex : 20-8-18E). Then, send a DMC_SET_PAR_B_SPEC to really send the parameters to the DEC. Note: with this command, we set bit1 to 1 (activate CRE). Bit0 has been set to 1 by the DMC_SWON_B_SPEC command but, bit0 is not commandable through the DMC_WRT_B_SPEC_PAR. The only way to switch on/off the detector array is to use the trigger commands.
7. Then, you should set the bias voltages by writing the first 4 or 5 parameters of the block. Make sure to copy the latest values you use for the first 3 parameters.

#### 4.4.4.2 CRE bias commanding

The schematic below explains the meaning of R Bias, D Bias commands and shows how the Bias housekeeping is acquired.




#### 4.4.4.3 CRE housekeeping

CRE hk is related to GND (not to VSS as in IMEC ICD)

#### 4.4.4.4 Heater and Flasher

To control heater and flasher, the DEC must be switched ON. Then, 12 trigger commands are used to switch them on/off and to set their currents. Nothing is done through write commands.

Each heater and flasher must be switched on separately through one of these commands:
DMC_SWON_BD_HEATER
DMC_SWON_BD_FLASHER
DMC_SWON_RD_HEATER
DMC_SWON_RD_FLASHER
When switching on the heater or flasher, its current is automatically set to zero.

Then, you should set the current in each of the heater or flasher through one of these commands:
DMC_SET_B_SPEC_HEATER_C

DMC_SET_B_SPEC_FLASHER_C
DMC_SET_R_SPEC_HEATER_C
DMC_SET_R_SPEC_FLASHER_C

Then, each heater and flasher must be switched off separately through one of these commands:
DMC_SWOFF_BD_HEATER
DMC_SWOFF_BD_FLASHER
DMC_SWOFF_RD_HEATER
DMC_SWOFF_RD_FLASHER
When switching off the heater or flasher, its current is automatically set to zero.

#### 4.4.4.5 CRE output conversion

The science packet contains measures of the CRE output voltage. The conversion is given by:
65535 corresponds to a span of 6.27 V at the CRE output (1 LSB = 94.91 µV).

If the CRE output is equal to VDDA + 0.727V, the measure gives 65535.
If the CRE output is equal to VSS + 0.627V, the measure gives 6553.
If the CRE output is below VSS + 0.627V, the measure is non linear (but we should never have measure within this range).

History:
The aim was to be able to measure the complete range between VSS and VDDA at the CRE output (span: 5.5V).
The sensitivity of the converter itself is:
0 corresponds to 0V at the converter input
65535 corresponds to 5V at the converter input

The input amplifier in front of the converter can not reach 0V at its output; the first half volt can be non linear.

The gain of the input amplifier has been set to 26.7/33.2 (0.804) in such a way a span of 5.597V at the CRE output corresponds to a span of 4.5V at the converter input (between 0.5V and 5V)

Originally,
If the CRE output was equal to VDDA, the measure gave 65535.
If the CRE output was equal to VSS, the measure gave 6553.

Following a request of MPE to be able to measure a CRE output a bit higher than VDDA:
If the CRE output is equal to VDDA + 0.727V, the measure gives 65535.

If the CRE output is equal to VSS + 0.627V, the measure gives 6553.

#### 4.4.4.6 Summary of commands and telemetry

**Trigger Commands:**

- 12 DMC_SWON_B_DEC
- 13 DMC_SWOF_B_DEC
- 14 DMC_SWON_B_SPEC
- 15 DMC_SWOF_B_SPEC
- 16 DMC_SET_PAR_B_SPEC
- 17 DMC_SET_B_SPEC_HEAT_C
- 18 DMC_SET_B_SPEC_FLASH_C
- 19 DMC_SWON_R_DEC
- 20 DMC_SWOF_R_DEC
- 21 DMC_SWON_R_SPEC
- 22 DMC_SWOF_R_SPEC
- 23 DMC_SET_PAR_R_SPEC
- 24 DMC_SET_PAR_BOTH_SPEC
- 34 DMC_SET_R_SPEC_HEAT_C
- 35 DMC_SET_R_SPEC_FLASH_C
- 60 DMC_SWON_BD_HEATER
- 61 DMC_SWOF_BD_HEATER
- 62 DMC_SWON_BD_FLASHER
- 63 DMC_SWOF_BD_FLASHER
- 79 DMC_SWON_RD_HEATER
- 80 DMC_SWOF_RD_HEATER
- 81 DMC_SWON_RD_FLASHER
- 82 DMC_SWOF_RD_FLASHER

**Write Commands:**

- 151 DMC_WRT_B_DEC_REC_OPT
- 152 DMC_WRT_R_DEC_REC_OPT
- 154 DMC_WRT_R_SPEC_PAR
- 155 DMC_WRT_B_SPEC_PAR
- 158 DMC_WRT_B_PACKT_ENC_LINK
- 159 DMC_WRT_R_PACKT_ENC_LINK

**HK nominal:**

- 265-400 copy of HK from DEC

- 200 DMC_DECB_REC_STA
- 201 DMC_DECB_CTRL_ST
- 202 DMC_BLUE_PAC_ENC
- 203 DMC_DECR_REC_STA
- 204 DMC_DECR_CTRL_ST
- 205 DMC_RED_PAC_ENC
- 228 DMC_DECB_REC_PAC
- 229 DMC_DECR_REC_PAC
- 230 DMC_DECB_CTRL_PA
- 231 DMC_DECR_CTRL_PA
- 232 DMC_BLUE_ENC_PAC
- 233 DMC_RED_ENC_PAC

**HK diag:**

- 579-618 copy of HK from DEC

### 4.4.5 Using the photometry detectors

To start using the photometry detectors:

1. Spacecraft should switch-on BOLC power
2. Establish the communication between DMC and BOLC (*DMC_RESET_SMCS_CHIP_2*)
3. Send commands to BOLC (*DMC_SEND_COMMAND_TO_BOLC*) to configure it
4. Start forwarding the data to SPU (set bit 2 of *DMC_WRT_BOL_REC_OPT* to 0)

#### 4.4.5.1 Summary of HLSW commands and telemetry

**Trigger Commands:**

- 33 DMC_SEND_COMMAND_TO_BOLC
- 89 DMC_RESET_SMCS_CHIP_2

**Write Commands:**

- 150 DMC_WRT_BOL_REC_OPT
- 158 DMC_WRT_B_PACKT_ENC_LINK
- 159 DMC_WRT_R_PACKT_ENC_LINK

**HK nominal:**

- 0-195 copy of HK from BOLC
- 206 DMC_BOL_REC_STAT

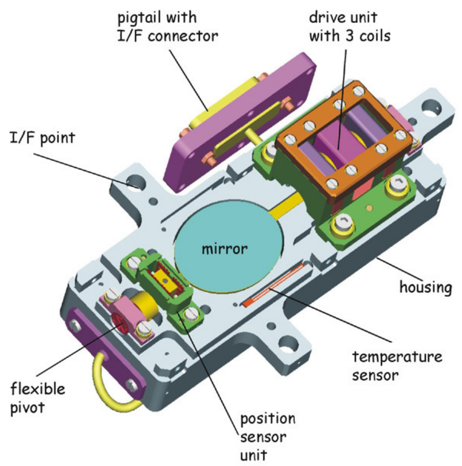
- 207 DMC_BOL_CTRL_STA
- 234 DMC_BOL_REC_PAC
- 235 DMC_BOL_CTRL_PAC
- 240 DMC_BOL_READ_CNT
- 449 DMC_BOLC_STATUS
- 481-508 copy of HK from BOLC

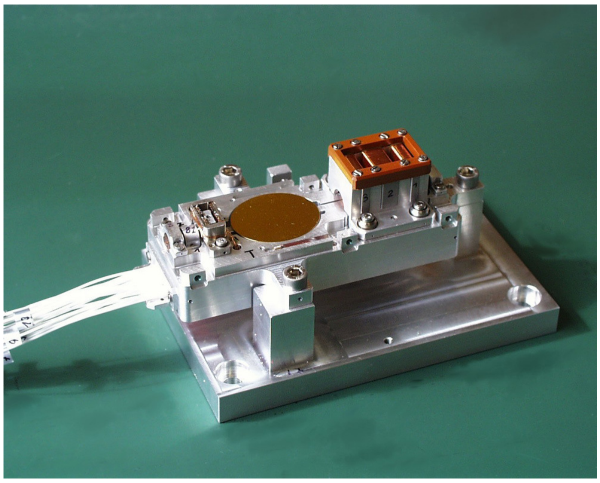
**HK diag:**

- none

### 4.4.6 The chopper : general description

Hereunder are given a schematic and a picture of the chopper, showing its main elements.



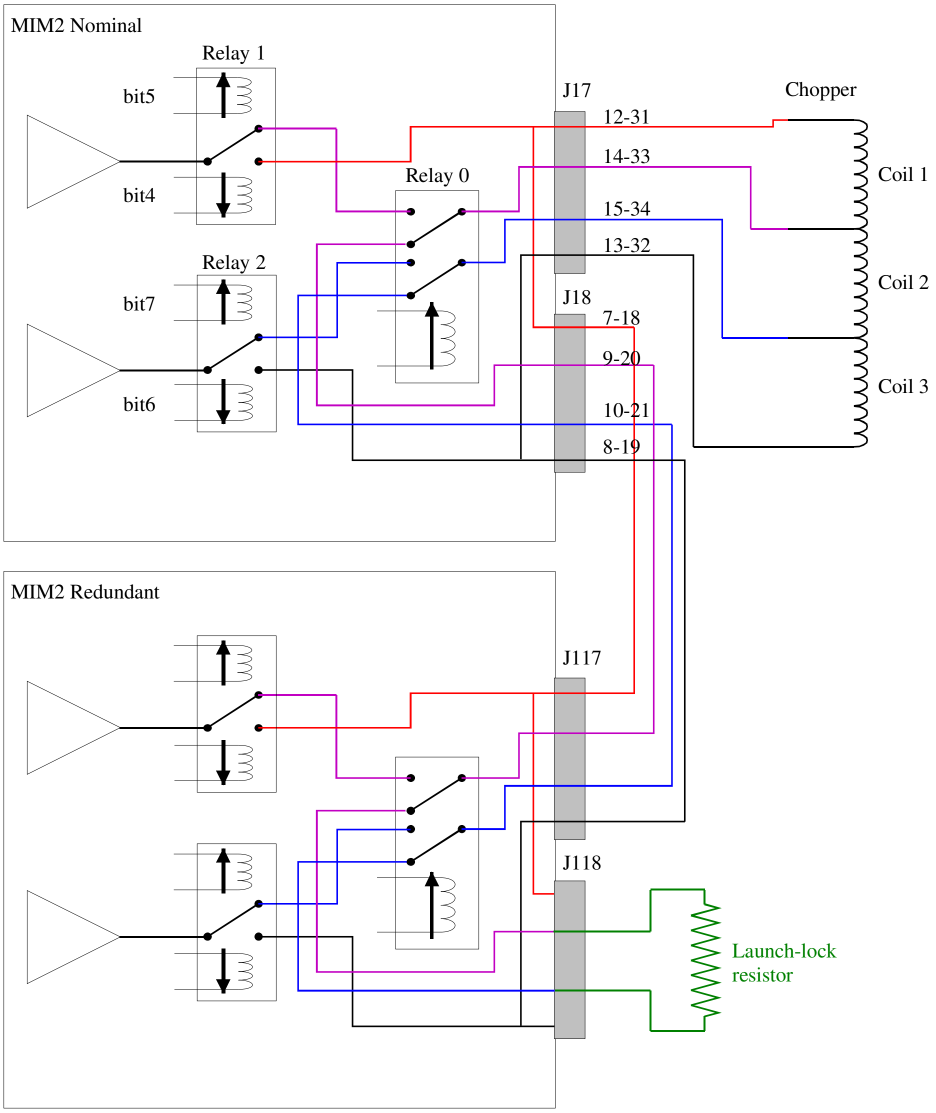
An important characteristic of the chopper is that the rotor (including the mirror) is mounted on spring bearings (named flexible pivots). This means that the chopper has an equilibrated rest position (at zero drive current) and that a constant current is required to maintain the chopper at a specified angle (as large the angle, as high the required current). Normally, the chopper should be mechanically aligned so that the rest position is corresponding to the zero optical position of the chopper.

#### 4.4.6.1 The chopper actuator

The chopper actuator is composed of three coils (1 big coil surrounded by two small coils) which can be independently commanded from the DMC using a set of relays. Indeed, there is no redundancy for the chopper actuator but all the three coils are connected to both nominal and redundant MIM boards and degraded mode operation (using one or two coils) can be used in case of problems.

The complexity lies in the fact that the chopper actuator is the only non-redundant device while everything else must be redundant, including the chopper control electronics. Therefore, the connection scheme shown in the figure below has been implemented in the DMC. On this figure, the default state (switch off state) of the relays is represented. In this configuration, coil 1 and coil 3 are disconnected from the electronics (red and black lines) and coil 2 is connected to a short-circuit resistor (used as a launch-lock) through the redundant MIM board (violet and blue lines).

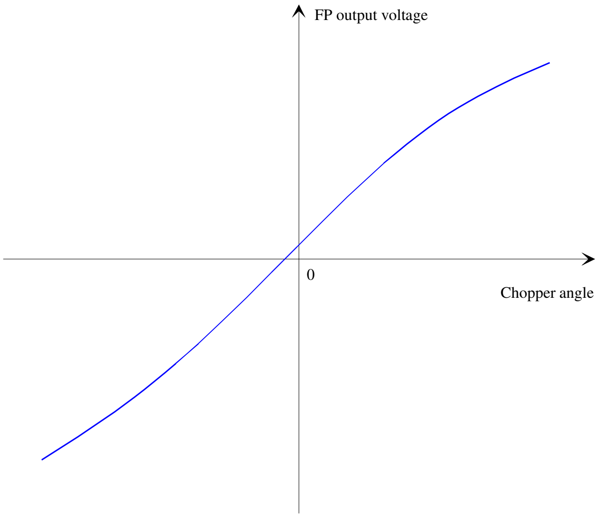
How the relays must be commanded to use the chopper will be explained in details in §4.4.7.1.



#### 4.4.6.2 Field plates position sensor

The chopper position sensor is a double differential magneto resistor named field plates. Two sensors are available, one connected to nominal DMC electronics (referred as FP1) and one connected to redundant electronics (referred as FP2).

Each sensor has a response curve w.r.t chopper angle shaping as shown in the figure hereunder. This curve is not linear and not symmetrical and may have an offset w.r.t the chopper zero (rest) position. Therefore, an accurate calibration curve is required to translate the readout voltage in angle units.

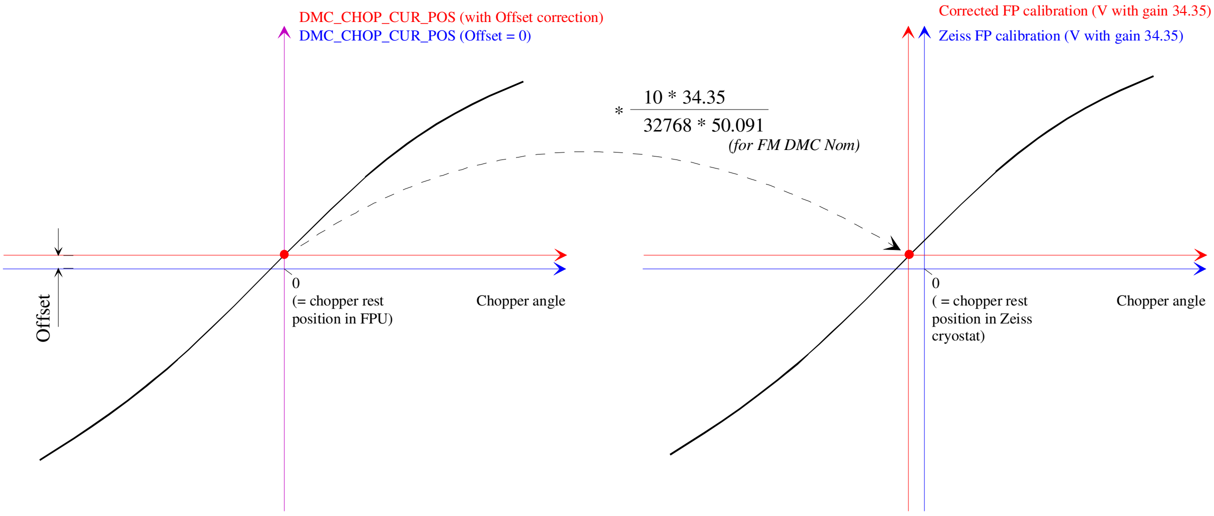



The FP signal is amplified before analog to digital conversion by a factor of 50 (actual gain value must be measured with accuracy on DMC hardware). The readout units of DMC_CHOP_CUR_POS are then in volts where ±10 V corresponds to ±32767 digits.

FP1 (nominal) amplification gain for DMC FM : 50.091

FP2 (redundant) amplification gain for DMC FM : 50.607

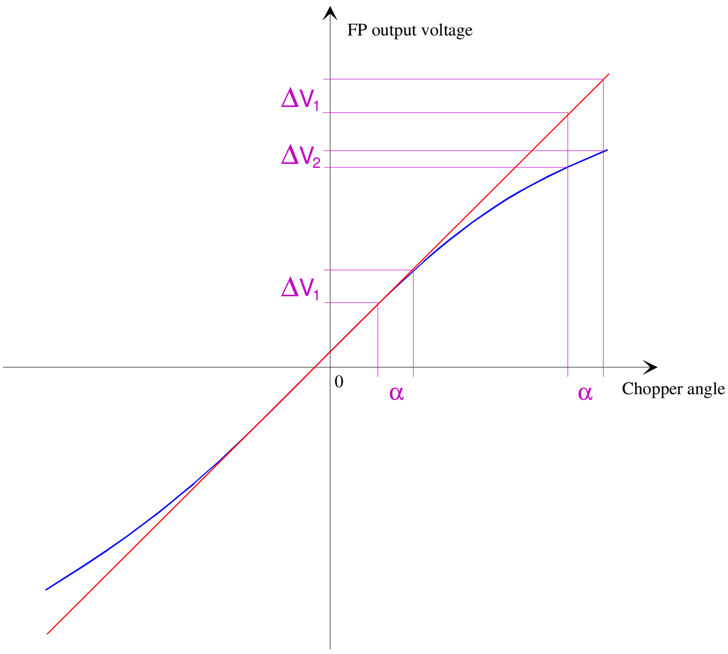
The Field plates calibration curves have been measured by Zeiss are are reported in chopper documentation (for the FM unit, see PACS-MA-TN-678). The measurement has been done in Zeiss test cryostat with an amplification of the FM signal by 34.35. The figure below shows how to use a Zeiss calibration curve and convert it in DMC units.




The process is as following :

1. Set chopper Offset parameter to 0
2. Measure DMC_CHOP_CUR_POS with controller Disabled and OFF to be sure that no current is flowing trough the coils. The chopper is then in its mechanical rest position, which by alignment is supposed to be the optical zero position in FPU.
3. Set Offset = - DMC_CHOP_CUR_POS (as measured in 2.)
4. Make the following calculation to find the zero position in Zeiss voltage units : Zero_Zeiss = -Offset*(10*34.35)/(32768*50.091) (50.091 valid for FM DMC nominal only)
5. Take this Zero_Zeiss point as the origin of the axes for defining a corrected calibration curve from the original documented by Zeiss.
6. Then conversion from DMC units to Corrected FP calibration units is done simply by using the scaling factor (10*34.35 FP1 (nominal) amplification gain for DMC FM : 50.091) (50.091 valid for FM DMC nominal only)

The FP signal is non-linear w.r.t chopper angular deflection and asymetric w.r.t the neutral position. Therefore, a same angular increment will lead to different FP readout amplitude depending of the position of the chopper. This is shown in the figure hereunder where the FP response curve is compared with an ideal linear characteristic (fitted to the FP curve for low angles).



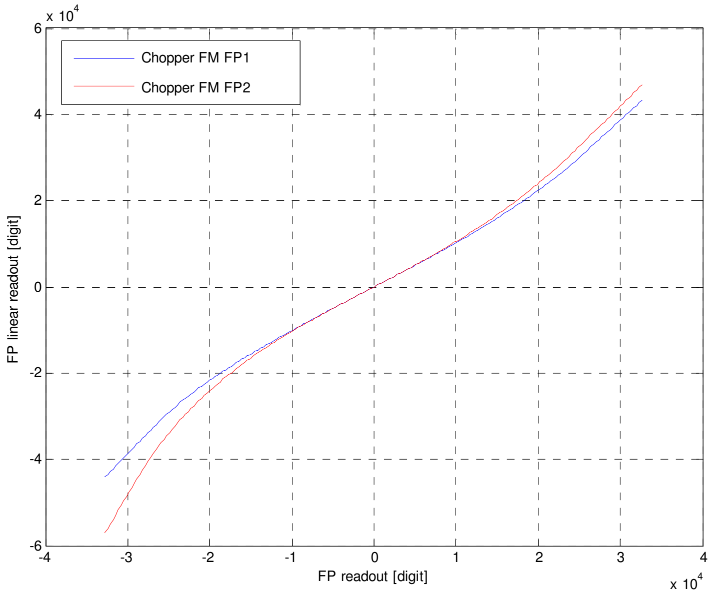

This change of FP sensitivity w.r.t the chopper position induces that a given controller parameters set will produce different performance depending on the chopper position. Especially, parameters optimised for the nominal operating range (±4.1°) will not be optimum for larger angles (~±9°) used to aim the calibration sources. Also, the asymmetry of the FP response curve induces that the controller performance will be different for positive or negative deflection and therefore, optimisation of parameters is always a compromise.

The OBSW can use the FP calibration curve to linearise the position readout before to be used by the controller. A virtual linear response curve is used, having the same sensitiviry than the FP curve for the small angles (like shown in red in the figure above). In order to convert the FP readout to a virtually linear readout, a conversion look up table is used. The conversion is internal to the controller software and is not applied to the FP readout as seen in the Housekeeping (DMC_CHOP_CUR_POS).

The parameter SelectFieldPlateLUT allows selecting between the following FP readout conversions :

| | |
|---|---|
| SelectFieldPlateLUT = 0 | Look up table for linearisation of FP1 |
| SelectFieldPlateLUT = 1 | Look up table for linearisation of FP2 |
| SelectFieldPlateLUT ≥ 2 | No conversion |

The figure below is a plot of the look up tables used for FP1 and FP2 of the chopper FM.




### 4.4.7 Using the chopper

#### 4.4.7.1 Selecting chopper actuator coils

For an understanding of the following explanation, refer to the figure presented above in §4.4.6.1.

The driving amplifier is connected to the chopper coils using a set of 3 relays. One relay (relay 0) has a stable power off state (which is as shown in the figure) and its coil must be continuously powered to switch to the other position. The two other relays (relay 1 and 2) are bi-stable relays, meaning that no power is required to keep them in any position and a power pulse must sent to the correct relay coil to switch from one stable position to the other. Therefore, they have no default power off position by hardware but the software command them in their default position (as represented in the figure) when power off of the chopper is commanded using *DMC_SWOF_CHOP_CONT*. It is important to note that all the relays of the non used nominal or redundant electronics must be in their default position. If it is not the case, the currently used electronics will not be able to command the chopper properly. Therefore, a correct switch off of the chopper by commanding *DMC_SWOF_CHOP_CONT* must be done before switching from nominal to redundant electronics or inversely.

At switch on of the chopper controller using *DMC_SWON_CHOP_CONT*, the relay 0 is powered which disconnect the chopper from the launch-lock resistor. It stays powered until the switch off command *DMC_SWOF_CHOP_CONT* is sent.

The other two bi-stable relays can then be commanded using the *DMC_SET_CHOP_COIL_DRIVE* command with a parameter in which bit 4 to 7 define the position of the relays. In the figure, the arrow indicates the direction of activation when the corresponding bit is set to 1. There are 4 different configurations as shown in the table.

| Chopper coils used | bit 7 | bit 6 | bit 5 | bit 4 | Parameter value |
|---|---|---|---|---|---|
| Coils 1-2-3 (nominal) | 0 | 1 | 0 | 1 | 0x28 |
| Coils 1-2 (bypass 3) | 1 | 0 | 0 | 1 | 0x48 |
| Coils 2-3 (bypass 1) | 0 | 1 | 1 | 0 | 0x30 |
| Coil 2 (bypass 1&3) | 1 | 0 | 1 | 0 | 0x50 |

At switch on of the chopper controller (*DMC_SWON_CHOP_CONT*), the nominal configuration is automatically selected. In case the nominal configuration is used, it is therefore not necessary to use *DMC_SET_CHOP_COIL_DRIVE* command with parameter 0x28 after a switch on.

Note that changing the chopper coils configuration can be done only when the controller is disabled and that chopper controller parameters must be adapted for each configuration.

#### 4.4.7.2 Nominal mode operation

To start using the chopper, one should :

1. Switch-on the chopper controller (*DMC_SWON_CHOP_CONT*)
2. If not operating in nominal coils configuration (3 coils used), change the coils configuration (*DMC_SET_CHOP_COIL_DRIVE) and update controller parameters*
3. Enable the chopper controller (*DMC_ENABLE_CHOP_CONT*)
4. Then, any of the 4 move commands can be sent
5. When done, disable the chopper controller (*DMC_DISABLE_CHOP_CONT*)
6. Switch-off the chopper controller (*DMC_SWOF_CHOP_CONT*)

#### 4.4.7.3 Changing the controller parameters

The chopper parameters are defined by Zeiss for the different coils configurations and temperature conditions. The way to translate the Zeiss parameters in DMC units is detailed in §4.2.1.4.

You should only use those provided controller parameters. Setting a wrong value in any of these parameters could damage the chopper or its driver electronics.

An important parameter is the OutputLimit parameter which defines the maximum output current which can be commanded to the chopper coils. Indeed, the driving amplifier is designed to output a maximum current of ±133 mA (when DMC_CHOP_OUTPUT = ±32767). This current is required in degraded mode when using only one coil in order to reach the extreme angle deflection of the chopper. However, in nominal (three coils) or two coils configuration, the current must be limited to the appropriate value to avoid any over travel outside of the operational range of the chopper and damage while knocking the hard stops.

Another important parameter is the PosLimit parameters which defines a maximum deflexion of the chopper above which the controller will be disabled. This must be set slightly above the operational range in order that any non nominal event which would drive the chopper at an angle higher than the nominal operational range will trigger the disabling of the controller while passing through the PosLimit value.

#### 4.4.7.4 Open loop mode operation

The chopper can be operated in open-loop mode. That means that a specified current can be applied to the actuator directly.

When commanding the chopper in open loop, there is of course no damping of the natural vibration frequency of the chopper (the mass-spring frequency of rotor on flex pivots is about 30 Hz). Therefore, if a current step is applied, the chopper will oscillate for 5-10 seconds before to stabilise. To avoid excitation of the chopper oscillation frequency, the current can be applied slowly by using a very small Rate parameter. In that way, the current will be applied using a low frequency sinus function until the specified target is reached.

Selecting the open-loop mode is done via the command (*DMC_SET_CHOP_COIL_DRIVE*) with bit 8 set to 1 (If changing only the open/close loop mode without changing the coils configuration, bits 0-7 can be set to zero). To go back to closed loop mode, send this command with bit8 set to 0. This must be done when chopper controller is switched-on and when the chopper controller is disabled.

Then any of the 4 move commands can be sent but the parameter is now the driving current rather than the angle setpoint. The units are in mA where ±32767 = ±133 mA.

#### 4.4.7.5 Summary of commands and telemetry

**Trigger Commands:**

- 49 DMC_SWON_CHOP_CONT
- 50 DMC_SWOFF_CHOP_CONT
- 51 DMC_ENABLE_CHOP_CONT
- 52 DMC_DISABLE_CHOP_CONT
- 53 DMC_MOVE_CHOP_ABS
- 54 DMC_MOVE_CHOP_REL
- 55 DMC_MOVE_CHOP_ABS_DITHER
- 56 DMC_MOVE_CHOP_REL_DITHER
- 57 DMC_SET_CHOP_COIL_DRIVE

**Write Commands:**

- 144 DMC_WRT_CHOP_CONF_PAR
- 153 DMC_WRT_MAX_DITHER

**HK nominal:**

- 209 DMC_CHOP_CTRL_ST

- 244 DMC_CHOP_CUR_POS
- 245 DMC_CHOP_SETPOIN
- 246 DMC_CHOP_TARGET
- 247 DMC_CHOP_PID_ERR
- 248 DMC_CHOP_PID_ACC
- 249 DMC_CHOP_MAX_DIT
- 258 DMC_CHOP_OUTPUT

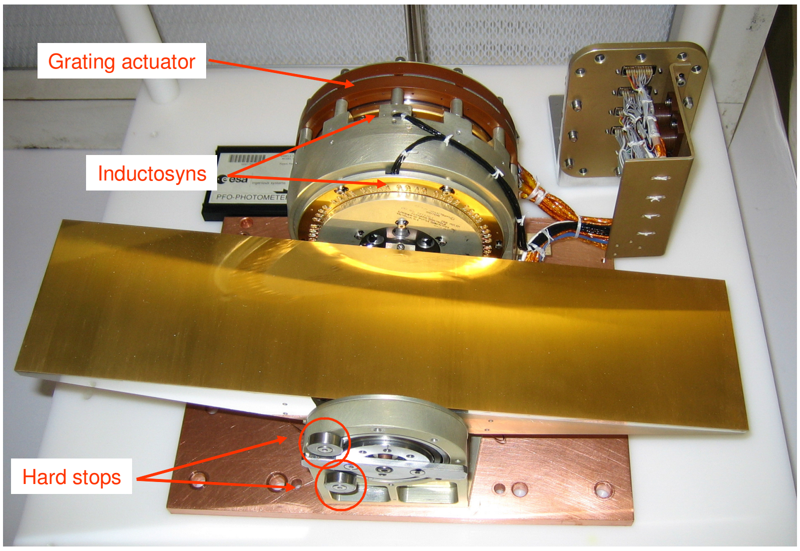
**HK diag:**

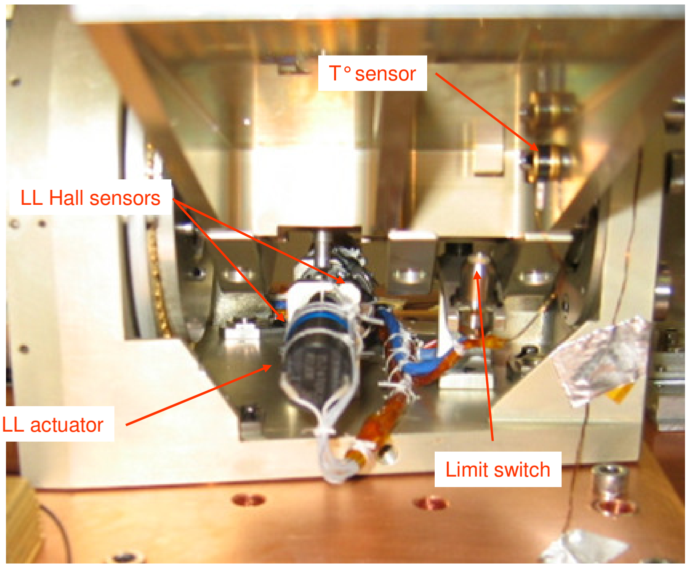
- 557 DMC_CHOP_VA
- 561 DMC_CHOP_IA
- 565 DMC_CHOP_VB

### 4.4.8 The Grating : general description

The two pictures hereunder show the grating mechanism and the most interesting elements which are used for its control, i.e. :

- The grating main actuator
- The inductosyn position sensor
- The hard stops
- The launch-lock mechanism
- The temperature sensors



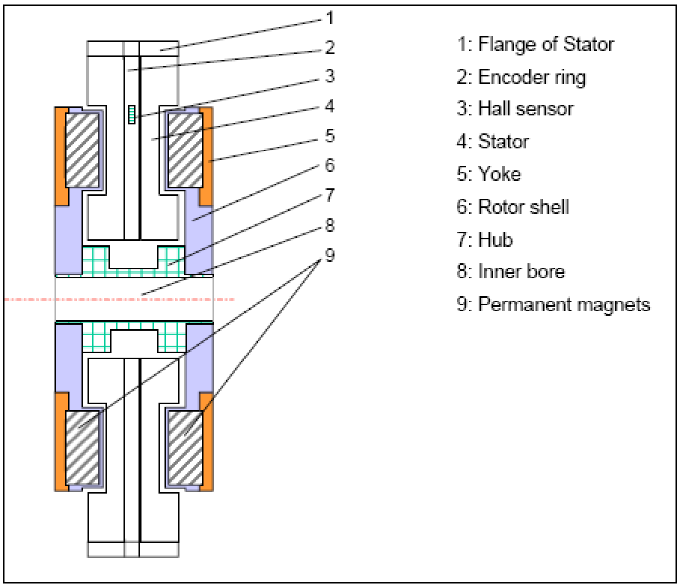



#### 4.4.8.1 The grating actuator

A detailed description of the grating actuator can be found in document PACS/GA-SP-001-TTL, which is included in the Grating PFM end item data package PACS-CL-DP-004.

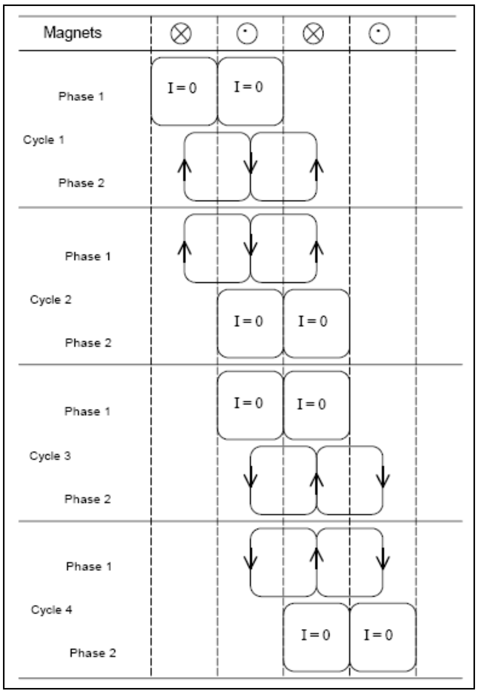
There are two electrical circuits in the actuator, one nominal and one redundant, each composed of one pair of coils and two hall sensors.

The actuator is composed of a turning rotor part and a fixed stator part (see figure below for description).



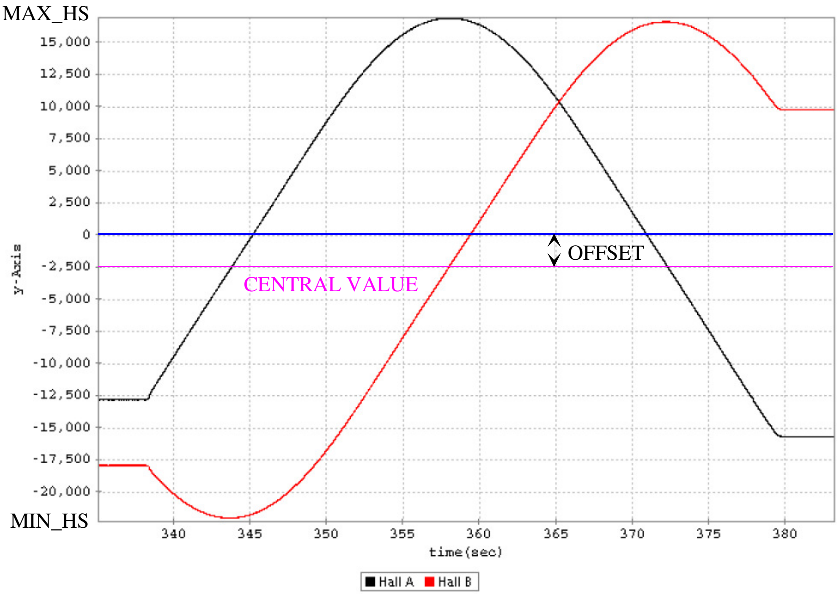
The rotor part is composed of 12 pair of magnets uniformly distributed on the circumference (i.e. one pair of magnets each 30°), and with alternating magnetic field. The stator part is composed of two coils (phase 1 and phase 2) placed between the magnets (in the magnetic field) and dephased of 15° (i.e. 1/4 of period) one to the other.

To turn the rotor, the current in the coils must be commanded as shown in the sketch hereunder.




This means that to turn the actuator continuously, sine and cosine current modulation must be sent in each coil, with an angular period of 60°. This is done using two hall sensors which are aligned with the coils and which have the property to output a signal proportional to the magnetic field. Therefore, by reading the two hall sensors, the sine and cosine distribution of the current in the coils can be determined.

As the total grating angular range is about 40°, only 2/3 of a period of the hall sensors can be measured while moving the grating from one hard stop to the other. An example off hall sensors signals on the whole grating range is shown in the figure hereunder.




In order to make the current repartition correctly, the software requires that the hall sensors signals are sinusoids with an amplitude of 65535 (i.e. between -32767 to 32767). As shown in the figure, this is of course not the case and also the amplitude of the signals is changing with temperature. Therefore, a scaling parameter is used which is defined as following (refer to the picture) :

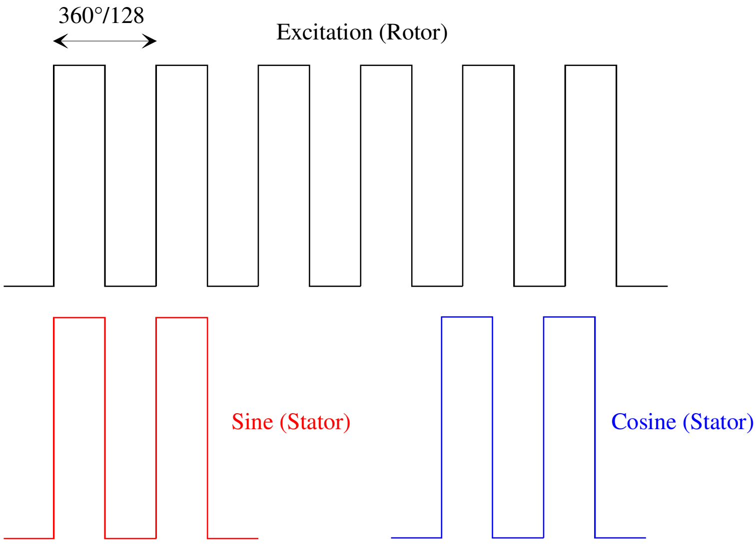
SCALING = 1024*65535/(MAX_HS-MIN_HS)

At ambient temperature, the hall sensors signals are symmetrical around 0. However, at cold temperature, an offset appears as it is the case in the figure (meaning that the signal is different from 0 even if there is no magnetic field). An offset parameter is then also used to recover symmetrical signals before to be used by the software. It can be set using the command *DMC_WRT_GRAT_HALL_OFFSET*.

See §4.2.1.2 to see how are defined and how to change these parameters.

#### 4.4.8.2 The inductosyn position sensor

The inductosyn position sensor is an inductive sensor composed of two facing disks, one fixed (stator) and one turning with the grating (rotor). On the rotor disk, there is one printed circuit shaped like a square wave and used as an excitation coil, with 128 periods on the total circumference. On the stator disk, there are two similar printed circuits used as secondary coils and dephased of 1/2 period one to the other (see the picture below). The excitation signal (and therefore the two readout signals) is a sine with a frequency of 16384 Hz. The two readout signals are in phase with each other but when the grating is turning, their amplitudes are varying just like sine and cosine functions with a period corresponding to an angle of 360°/128.




From the sine and cosine readout signals, the electronics output a 16 bits value which is an absolute position reading within a period of the printed circuit. There is an additional counter which increment or decrement when there is a transition of the position readout from 0xFFFF to 0x0 or from 0x0 to 0xFFFF respectively. This period counter is an 8 bits integer. Therefore, the position readout is a 24 bits signed integer.

At switch on of the grating controller using the command *DMC_SWOF_GRAT_CONT*, the grating is in an arbitrary position and the period counter must be reset by placing the grating in a well known reference position. This is the reason why a homing process is required and two hard stops at extreme positions of the grating are used as reference.

#### 4.4.8.3 The launch-lock

The launch-lock is composed of two actuators (one nominal and one redundant) which are each connected to one DMC MIM3 board (respectively to nominal and redundant MIM3). However, to increase the reliability and decrease the required driving current, the two actuators can be operated together, whatever the nominal or the redundant electronics is in use. Each actuator is driven with a maximum current of 400 mA.

The launch-lock is equipped with hall sensors indicating the unlocked and locked positions. In order to minimise the dissipated power, these hall sensors are powered only when the actuators are driven. Therefore, the reading of the launch-lock position is only possible during a launch-lock activation operation.

### 4.4.9 Using the grating

The grating and the filter wheels are driven by the same extension board and power supply. So, only one of these mechanisms can be used at a time. Switching-on one of these mechanism controllers will automatically switch-off the one that is currently on.

There is only one command to switch-off the controller that is currently on : *DMC_SWOF_GRAT_CONT.*

The grating position encoder (inductosyn) will be powered-off only when the *DMC_SWOF_GRAT_CONT* is received. By this way, it is possible to move the grating, then a filter wheel and then the grating again without loosing the grating position. This can be achieved if you use only *DMC_SWON_FW_SPEC, DM_SWON_FW_PHOTO* and *DMC_SWON_GRAT_CONT* to toggle between the mechanism controllers and never use *DMC_SWOF_GRAT_CONT* (see section 4.4.11 for details).

It is recommended to disable de grating controller before any switching from the grating to one of the filter wheels. This is done by sending the command *DMC_DISABLE_GRAT_CONT.*

#### 4.4.9.1 Hardware settings for position readout

The inductosyn readout electronics needs a very accurate tuning of its parameters (on MIM1 and MIM3 boards) depending on which inductosyn sensor is connected and in which temperature conditions it is used. The DMC is adjusted to operate the grating at cold conditions and with nominal MIM boards connected to nominal inductosyn and redundant MIM boards connected to redundant inductosyn.

In all other conditions, the inductosyn signal may be very noisy and not accurate. That means in particular that the grating cannot be operated in closed loop at ambient conditions, i.e. the grating controller cannot be enabled. Command *DMC_ENABLE_GRAT_CONT* is therefore only allowed in cold conditions. However, open loop mode can be used in any conditions.

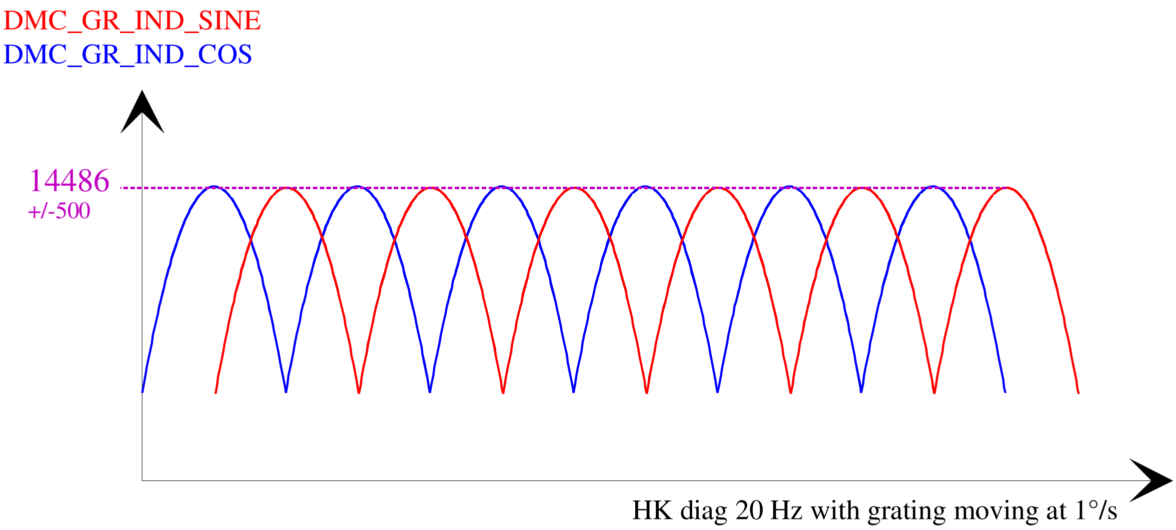
The MIM3 hardware settings are corresponding to an adjustment of the amplitude of the excitation and hence of the sine and cosine readout signals. The MIM1 hardware settings are corresponding to an adjustment of the phase of sine and cosine signals with a reference signal, which is required for the converter A/D (AD2S80) to operate correctly. The MIM1 settings are realised by adjusting the value of fixed resistors while a software command allows some adjustment of the amplitude of the excitation.

The amplitude of the excitation can be adapted by using the command *DMC_WRT_GRAT_INDUCT_AMPL* with a parameter defining the amplitude of the excitation. This parameter is a 12 bits integer and therefore, the maximum sine and cosine amplitude will be realised with an amplitude command of 4095.

Two HK diagnostic variables DMC_GR_IND_SINE and DMC_GR_IND_COS allow having a readout of the sine and cosine amplitude. The amplitude of the excitation must be adjusted to get a maximum amplitude of sine and cosine signals of 2 Vrms (±10 %). This is done as following :

- Start HK diag at 20 Hz for DMC_GR_IND_SINE and DMC_GR_IND_COS
- Move the grating relative by 0x30000 at 1°/sec. (rate = 3)

- Analyse HK diag records. Figure hereunder shows how it looks like. The goal of the process is to obtain a maximum amplitude for these signals of 14486 (±500) digits. If the maximum amplitude is different, adjust the excitation amplitude and restart the measurement. This must be repeated until the target of 14486 is reached.




<u>ATTENTION : this process is done by operating the grating in closed loop. It is therefore only allowed when the inductosyn amplitude is already set to a value not too far from the target, i.e. for some periodical adjustment if required. If it is not the case, this will lead to unstability of the controller and open loop mode must be used to move the grating for recording HK diag data. In any case, changing the amplitude of the inductosyn signals must be done carefully and by experienced users only.</u>

<u>Also note that, since the sine/cosine measure circuit is using some spare channels of the calibrations sources circuit, they can not be measured while the calibration sources are operated. Both calibration sources must be switched off to be able to make this measure.</u>

#### 4.4.9.2 Changing the controller parameters

You should only use controller parameters provided by CSL. Setting a wrong value in any of these parameters could damage the grating or its driver electronics.

#### 4.4.9.3 Operating the launch-lock

The nominal operating mode is by driving the two launch-lock actuators simultaneously. Commanding only one actuator must be used only for testing or for degraded mode operation if required.

When launch-lock activation is commanded, the driving current is applied during about 5 s, regardless the value of the position sensors. The launch-lock position sensors are powered on only when the launch-lock motor is operated. Since the HK period is 2 seconds, the status bit of the position sensors will appear only in 2 or 3 HK packets.

Unlocking:
You should first switch-on the grating controller *DMC_SWON_GRAT_CONT*. Then send the *DMC_UNLOCK_GRAT* with parameter 28 (for the open-loop mode with two actuators). Then, the grating can be used as usual.

Locking:
The grating controller should be powered-on but disabled. Before disabling the grating controller, you should first move the grating to a central position (~500000). Then, Send the *DMC_LOCK_GRAT* command with parameter 12 (for the open-loop with two actuators).

Note that the launch-lock can be operated nominally at any temperature conditions, assuming that the grating is in horizontal position.

#### 4.4.9.4 Grating position readout

As explained in §4.4.8.2, there are 128 periods for for 360° with a position readout on 16 bits in one period. Therefore conversion from DMC_GRAT_CUR_POS to angle is made as following :

Angle [deg.] = DMC_GRAT_CUR_POS*360/(128*65535)
(For exemple DMC_GRAT_CUR_POS = 23301 corresponds to 1°)

In degraded mode, the grating position is given by DMC_GR_DEG_POS (available in diagnostic hk only). This position is actually the index in the sine array that is used to generate the command of the grating (see §4.4.9.10 for details about the degraded mode). This value is updated only when you move and when you switch-on the controller. This value is not reflecting the actual position but the command that is applied to the grating; this means that if the grating oscillates, you will not see it here, if it is not at the commanded position, you will not see it here as well (the only information about the real position of the grating can be retrieved from the hall sensors).

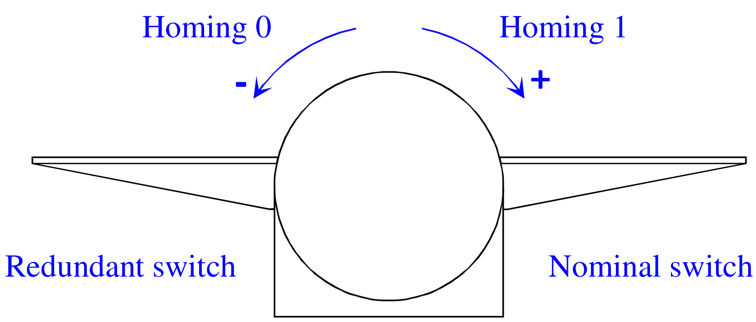
#### 4.4.9.5 Changing the inductosyn sign

From version 6.000, this command has been disabled since both FM inductosyn have the same orientation. Version 6.000 is not compatible with the redundant circuit of the QM grating !

#### 4.4.9.6 Homing the grating

As explained in §4.4.9.4, the period counter of the inductosyn must be reset with grating being positioned at one of the two hard stops. Just before reaching the hard stop, a limit switch is pressed to asses that the grating is well blocked against the hard stop and not inside its operational range due to an eventual problem. The limit switch position can be found in the HK in DMC_GRAT_CTRL_ST variable (bit 23), see §5.3.2.

The schematic below shows the position of the two hard stops and limit switches related to the inductosyn position direction.




**View of the grating when facing the main actuator**

Two different homings can be done by sending the homing command *DMC_HOME_GRAT* with parameter 0 or 1 (see §4.1), respectively using the hard stop at zero and full range positions. The period counter is reset in such a way that an absolute position readout after a homing process is independent of the used homing command. Here is an example to explain how the inductosyn homing is done for each different homing (numbers are given for example only and are not related to any real situation) :

*Consider that parameter RANGE = 0x00100000 and X is an arbitrary hexadecimal value (0…F)*

| | ***Position at switch on*** | ***Position at hard stop (zero side)*** | ***Position at hard stop after homing*** *⇒ Period counter set to 0* | ***Position at other hard stop (full range side) after homing*** |
|---|---|---|---|---|
| ***Homing 0*** | *0xXXXXXXXX* | *0xXXXX5DF5* | *0x00005D5F* | *0x001050C7* |

| | ***Position at switch on*** | ***Position at hard stop (full range side)*** | ***Position at hard stop after homing*** *⇒ Period counter set to RANGE* | ***Position at other hard stop (zero side) after homing*** |
|---|---|---|---|---|
| ***Homing 1*** | *0xXXXXXXXX* | *0xXXXX50C7* | *0x001050C7* | *0x00005D5F* |

However, each limit switch is connected to a given electronics, i.e. either to nominal or to redundant MIM boards. Therefore, the limit switch can be seen only at one of the two hard stops. The limit switch indication is available for nominal circuit when homing to 1 and for redundant circuit when homing to 0 (as shown on the sketch above).

#### 4.4.9.7 Determining the grating range parameter

This operation shall be repeated once for each model (and once for the nominal and once for the redundant).

Start HK diag monitoring the DMC_GR_IND_READ. Move the grating to both hardstops. From the diagnostic data, extract the min_pos and max_pos (minimum and maximum values of DMC_GR_IND_READ during the move).

The range is = (max_pos &[1] 0xFFFF0000) - (min_pos & 0xFFFF0000).

Note that the range for the Grating PFM is supplied by CSL and should nominally not be measured again. See §4.2.1.2.

#### 4.4.9.8 Nominal mode operation

To start using the grating, one should:

1. Switch-on the grating controller (*DMC_SWON_GRAT_CONT*). Note : this will switch-off the filter wheels controllers.
2. If necessary, unlock the grating (*DMC_UNLOCK_GRAT*)
3. If necessary, write the parameters blocks:
   - DMC_WRT_GRAT_CONF_PAR
   - DMC_WRT_GRAT_INDUCT_AMPL
   - DMC_WRT_GRAT_RANGE
   - DMC_WRT_GRAT_HALL_OFFSET
   - DMC_WRT_GRAT_CONF_FILT
4. Enable the grating controller (*DMC_ENABLE_GRAT_CONT*)
5. Home the grating (*DMC_HOME_GRAT*). This command takes around 60 seconds.
6. Then, any of the 2 move commands can be sent
7. When done, disable the grating controller (*DMC_DISABLE_GRAT_CONT*)
8. Switch-off the grating controller (*DMC_SWOF_GRAT_CONT*). Note : this command switches off the currently active controller (Grating or one of the filter wheel) and switches off the inductosyn power supply. So if you want to use the grating again, you will need to perform a *DMC_HOME_GRAT*.

#### 4.4.9.9 Using the grating in degraded mode

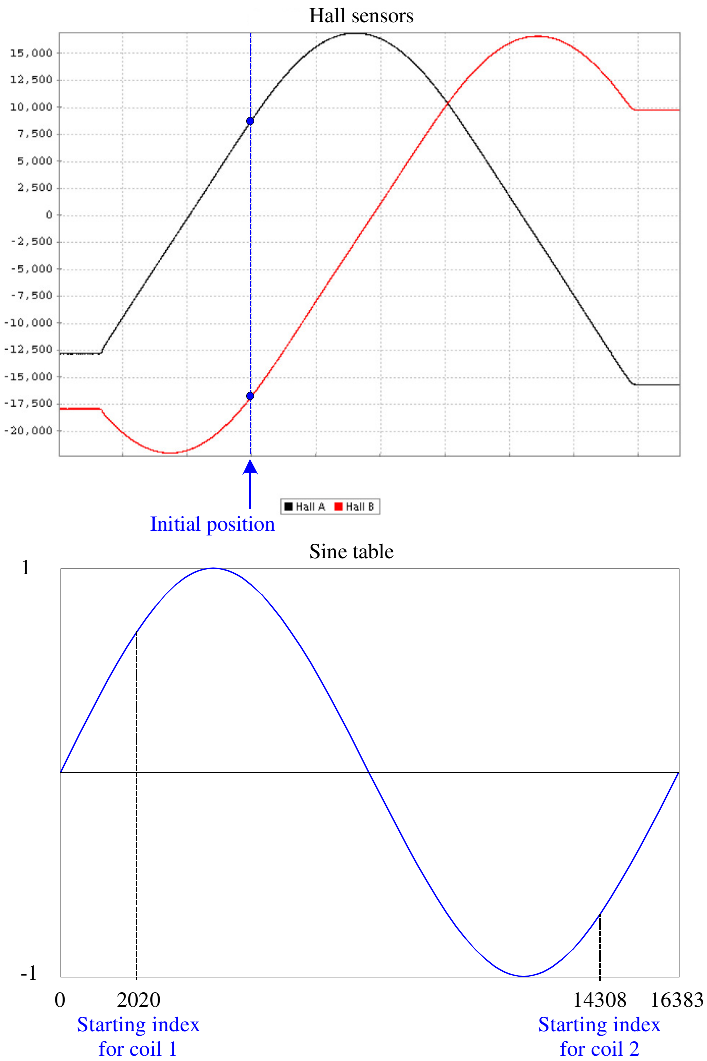
In the case of a failure of the inductosyn (and also for functional testing at ambient conditions), the grating can be used in degraded mode. This mode is an open loop mode (based on the operation of the filter wheels) : sine and cosine waveforms are sent in the motor coils. In this mode, the performance is lowered (settling time is longer, accuracy is lower, and stability is lower).

---

[1] & is the bitwise AND operator

To use the grating in degraded mode, you must enter the grating degraded mode (*DMC_ENTER_GRAT_CONT_DEG*). This must be done after switching on the grating (*DMC_SWON_GRAT_CONT*). Once you have switched on the grating, you must not enable the controller (in open-loop, the controller is inactive).

The software uses a sine table with 16384 entries for one sine period. When entering the degraded mode, a reading of the hall sensors is done to know the position of the rotor and therefore the starting position in the sine table to determine sine and cosine values of the current to be sent in the motor coils (see schematic hereunder). There is only one table, the index for the second coil being the index for the first coil dephased by a quarter of the table (4096).




Two different moves can then be commanded, using relative and absolute move commands. When commanding a relative move, the position specified is the relative index position w.r.t to the current index in the sine table (in the sketch above, a relative move of 2000 will move the grating from index 2020 to 4020). When commanding an absolute move, the position specified is the absolute index position in the table (in the sketch above, the same results than a relative move of 2000 can be obtained with an absolute move of 4020). At the end of a move, the starting index for the next move becomes the new current index. The current is maintained in the coils to hold the grating at the commanded position.

As already mentioned above, the value given in DMC_GR_DEG_POS is the current index in the sine table and is not obtained from any physical readout from the grating. One unit represents one step in the sine table. As the sine table has 16384 entries and as there are 6 periods for 1 turn, one units represents an angle of 360°/(6*16384) = 13,18 arcsec.

In order to determine the relation between a degraded mode position and a physical optical position of the grating, a calibration should be done to link the degraded mode position with an absolute inductosyn readout position. Actually, this can be done by plotting Hall sensors signals (DMC_FWGRAT_HALL_A and DMC_FWGRAT_HALL_B) w.r.t inductosyn position (DMC_GRAT_CUR_POS).

In open loop mode, the speed of the grating is determined by the degraded mode rate defined in DMC_WRT_GRAT_DEG_MODE_PARAM (that is not the same parameter as the nominal rate) . In this case, the degraded mode rate is the number of interrupt count between two steps in the sine table (for example, Rate = 32 => 32*16384/8192 = 64 sec for 60°). This will define the frequency of the sine and cosine driving functions sent to the coils of the actuator and therefore determine the speed.

#### 4.4.9.10 Working with the redundant grating

The grating is fully redundant and has two exactly identical electrical circuits for all its components. Therefore, there is no difference in operating the grating with redundant electronics than with nominal electronics.

The only difference will be in the optical calibration of the grating w.r.t its inductosyn position. Indeed, as the nominal and redundant inductosyns are physically different sensors, their mechanical alignment is not accurate enough to ensure that the same position readout is obtained for the same physical position of the grating. Actually, there will be an offset between the two inductosyns which must be measured by using a known physical reference. The hard stops can be used in that purpose; the offset being the difference in the position obtained using nominal and redundant inductosyns, with the grating against the hard stop after a homing has been done.

It is however recommended for better accuracy to make a wavelength calibration of the redundant circuit as well.

#### 4.4.9.11 Error detection

To prevent any damage to the grating mechanism or to its driver electronics, two protections have been implemented in the software.

Error limit:

If the error gets larger than the error limit, the controller is disabled and the output is set to zero. The grating will then be free running. In this case, an error is signalled in the grating controller status (but not in the software global status). Since the controller is disabled, it is no longer possible to move the grating until the controller is enabled again.

Power limit:

If the output of the controller is bigger than 100mA for more than 5 seconds, the controller will be disabled. This is to prevent warming of the driver electronics.

If you send the grating to or near the operating range limits, it will have to push the limit switch and its power consumption will increase. If the grating stays at this position for a too long period, this protection will trigger.

#### 4.4.9.12 Grating Health Check

The grating health check is done by acquiring 9 hk measures at high frequency :

- DMC_GRAT_CUR_POS
- DMC_GRAT_PID_ERR
- DMC_GRAT_OUTPUT
- DMC_FWGRAT_HALL_A
- DMC_FWGRAT_HALL_B
- DMC_FW_GR_VMOTA
- DMC_FW_GR_VMOTB
- DMC_FW_GR_IMOTA
- DMC_FW_GR_IMOTB

The acquisition shall be performed during a move from one hard stop to the other to cover the complete range. It shall first be performed on ground to set references and then at a TBD interval during flight. Values shall be compared to the reference to identify ageing problems.

The health check in cold conditions can be done in nominal operating mode. However, as the inductosyn signal at ambient is not nominal and as the controller cannot be enabled, the health check at ambient can be done only in degraded mode (open loop) operation.

To ease the interpretation of data, these charts shall be produced :

- Hall sensors signals related to inductosyn position (DMC_FWGRAT_HALL_A & DMC_FW_GRAT_HALL_B vs DMC_GRAT_CUR_POS)
- Error and commanded output current related to position (DMC_GRAT_PID_ERR & DMC_GRAT_OUTPUT vs DMC_GRAT_CUR_POS)
- Measured current and voltages in motor coils related to position (DMC_FW_GR_IMOTA & DMC_FW_GR_IMOTB & DMC_FW_GR_VMOTA & DMC_FW_GR_VMOTB vs DMC_GRAT_CUR_POS)

#### 4.4.9.13 Summary of commands and telemetry

**Trigger Commands:**

- 38 DMC_SWON_GRAT_CONT
- 39 DMC_SWOFF_GRAT_CONT
- 40 DMC_ENABLE_GRAT_CONT
- 41 DMC_DISABLE_GRAT_CONT
- 42 DMC_MOVE_GRAT_ABS
- 43 DMC_MOVE_GRAT_REL
- 44 DMC_MOVE_HOME_GRAT
- 45 DMC_ENTER_GRAT_CONT_DEG
- 46 DMC_EXIT_GRAT_CONT_DEG
- 47 DMC_LOCK_GRAT
- 48 DMC_UNLOCK_GRAT

**Write Commands:**

- 143 DMC_WRT_GRAT_CONF_PAR
- 160 DMC_WRT_GRAT_INDUCT_AMPL
- 161 DMC_WRT_GRAT_RANGE
- 162 DMC_WRT_GRAT_HALL_OFFSET
- 163 DMC_WRT_GRAT_DEG_MODE_PARAM
- 164 DMC_WRT_GRAT_CONF_FILT

**HK nominal:**

- 208 DMC_GRAT_CTRL_ST
- 250 DMC_GRAT_CUR_POS
- 251 DMC_GRAT_SETPOIN
- 252 DMC_GRAT_TARGET
- 253 DMC_GRAT_PID_ERR
- 254 DMC_CHOP_PID_ACC
- 256 DMC_FWGRAT_HALLA
- 257 DMC_FWGRAT_HALLB
- 452 DMC_GRAT_OUTPUT

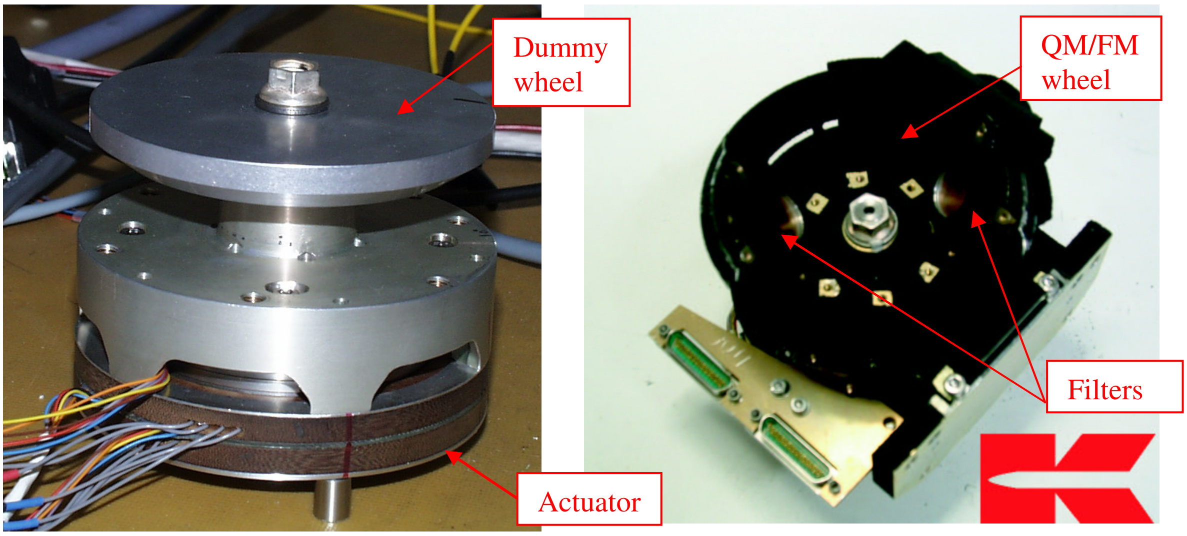
**HK diag:**

- 512 DMC_GR_IND_READ
- 513 DMC_GR_TURN_CAR
- 514 DMC_GR_PER_CAR
- 515 DMC_GR_DEG_POS
- 556 DMC_FW_GR_VMOTA

- 560 DMC_FW_GR_IMOTA
- 564 DMC_FW_GR_VMOTB
- 567 DMC_FW_GR_IMOTB
- 570 DMC_LL_CUR

### 4.4.10 The filter wheels : general description

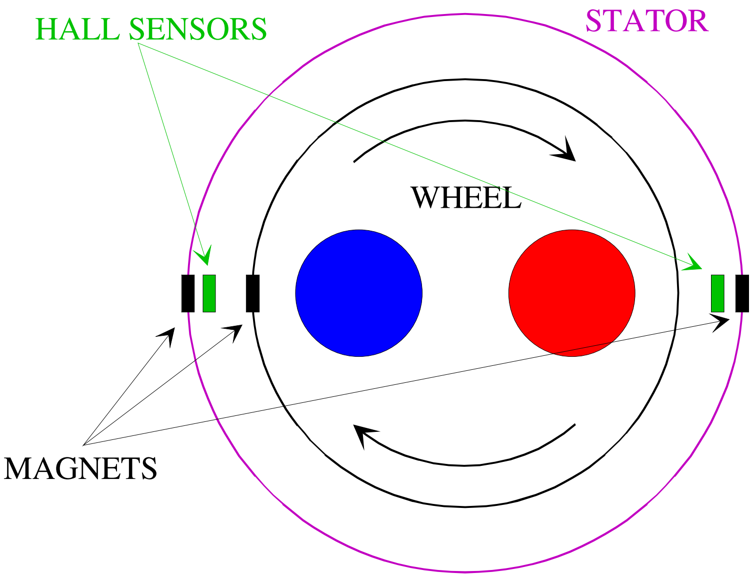
Hereunder are shown pictures of a dummy filter wheel (STM) and of one QM/FM model.



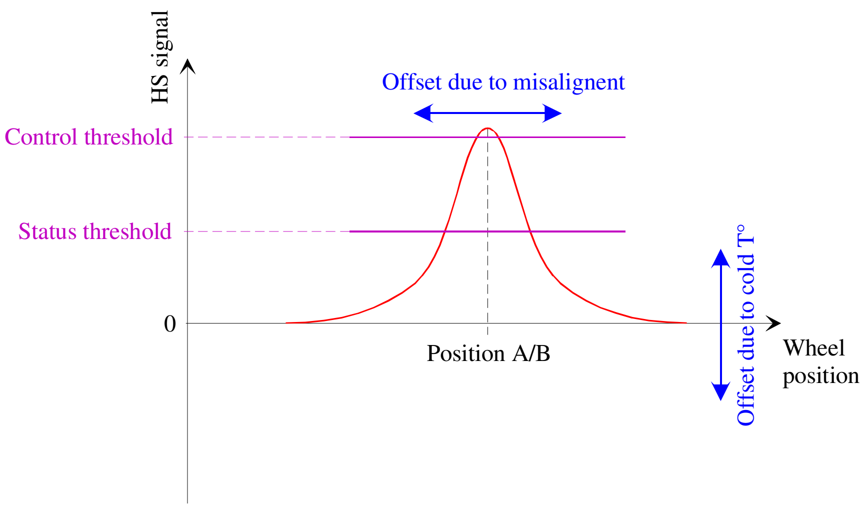

There are two filter wheels, one for the spectrometer and one for the photometer, but with identical design.

The actuator of the filter wheel is the same type as the actuator of the grating. It is built exactly like the grating actuator but with smaller size and different electrical characteristics. However, the operation of the filter wheel actuator is identical (see §4.4.8.1).

The filter wheel has two operating positions (corresponding to the two filters), which are locked by magnets as shown on the figure below. The magnets are also inducing a signal in the position indicator hall sensors by creating a magnetic field through the sensor when the wheel is in the right locked position.




The position hall sensor signal while the wheel is moving close to a locked position is like shown in the figure below. A control threshold parameter (see §4.2.1.3) is set for the software to determine when the filter wheel has reached the commanded position. That means that the current is reset to 0 as soon as the hall sensor signal is higher than the control threshold and then the wheel is free running and should be locked at right position by the magnet (see §4.4.11.1 for moving operations). If the magnet force is not sufficient to ensure positioning accuracy, the control threshold can be set close to the maximum of the hall sensor response to recover a good positioning accuracy and repeatablitity. A different status threshold parameter is used to switch the positioning status bit in DMC_FW_SPEC_CTRL / DMC_FW_PHOT_CTRL. This parameter must be set to a lower value in order to ensure that any back movement of the wheel when the current is reset will not produce an effect on the position status.



There are two effects which makes that the hall sensor signal is actually not as perfect as shown in the figure and this must be taken into account for adjusting the threshold parameter. First, the hall sensor can be not perfectly aligned with the magnet, inducing that the response curve will not be exactly centered on the locked position of the wheel. Secondly, the hall sensors have an offset at cold temperature, which can be either positive or negative depending of the sensor, inducing that the signal will not be 0 outside of the magnetic field. This will produce a vertical offset of the response curve.

Therefore, each hall sensor will have a different response curve and a pair of threshold parameters (status + control) is available for each position of the wheel (see §4.2.1.3).

### 4.4.11 Using the filter wheels

The filters wheels use the same driving electronics than the grating. The output of the driving amplifiers is switched between Grating, Spectro FW and Photo FW using a set of relays. Therefore, only one mechanism can also be commanded at a time.

For using a filter wheel, apply the following procedure :

1. If enabled, the grating controller must be disabled (*DMC_DISABLE_GRAT_CONT*).
2. Switch-on the spectro filter wheel controller (*DMC_SWON_FW_SPEC*) or the photo filter wheel controller (*DM_SWON_FW_PHOTO*). Note : this will activate the relays such that the output of the driving amplifiers are connected to the selected FW and therefore, the grating controller and the other filter wheel controller are switched off.
3. Then, the FW can be moved by any of the two moving commands (see §4.4.11.2).
4. Switch-off the controller (*DMC_SWOF_GRAT_CONT*). Note : this command switches off the currently active controller (Grating or one of the filter wheels)

#### 4.4.11.1 Moving a filter wheel

The filter wheels are operated by using the same principle than for the open loop (degraded) mode of the grating (see §4.4.9.9). There are however three small differences :

- At the end of a movement, the current output is set to zero and the filter wheel is maintained to its position only by the locking effect of the magnets if being at an operating position or is free running if positioned elsewhere.
- For that reason, the position of the filter wheel at the beginning of the next move is not exactly known. Therefore, a reading of the hall sensors (determining the starting index in the sine table) is made each time a move command is sent.
- The size of the sine table is different, one period corresponding to 256 (0x100) points. Therefore, a complete turn corresponds to a movement of 1536 (0x600) steps in the sine table.

There are two moving commands for the filter wheel, named closed loop or open loop commands. Using closed loop command (*DMC_MOVE_SPEC_FW_LOC* or *DMC_MOVE_PHOTO_FW_LOC*), the wheel is moved until the commanded position is reached (i.e. the hall sensor signal is higher than the specified control threshold). The wheel will turn in the commanded direction. If the control threshold value is never reached by the sensor signal (due to a defect or a wrong parameter setting), the wheel will make by default 1.5 turn.

Using open loop command (*DMC_MOVE_SPEC_FW_STEP* or *DMC_MOVE_PHOTO_FW_STEP*), the wheel is moved by a specified number of steps in the sine table, just like for the degraded mode operation of the grating. The direction of the wheel is determined by the sign of the command parameter. This mode can be used in case of a failure of a position hall sensor or if the wheel must be positioned to an intermediate position (not on a filter) for being used as a shutter in light path for example.

#### 4.4.11.2 Changing the filter wheel position between two acquisitions

Consider that the grating is powered on and that an acquisition sequence has ended. To change the filter wheel position now:

1. Disable the grating controller (*DMC_DISABLE_GRAT_CONT*)
2. Switch-on the spectro filter wheel controller (*DMC_SWON_FW_SPEC*). Note : this will switch-off the grating controller and the photo filter wheel controller but the grating position encoder will remain powered-on and will then continue reading the position.
3. Then, any of the 2 move commands can be sent
4. Then, switch-on the grating controller again (*DMC_SWON_GRAT_CONT*)
5. Enable the grating controller again (*DMC_ENABLE_GRAT_CONT*)
6. Start the new acquisition sequence.

#### 4.4.11.3 Filter wheels Health Check

To acquire all housekeeping related to the spectro filter wheel, start a diagnostic housekeeping acquisition with the following measures:

| HK ID (dec) | HK ID (hex) | HK ID |
|---|---|---|
| 210 | D2 | DMC_FW_SPEC_CTRL |
| 256 | 100 | DMC_FWGRAT_HALL_A |
| 257 | 101 | DMC_FWGRAT_HALL_B |
| 556 | 22C | DMC_FW_GR_VMOTA |
| 564 | 234 | DMC_FW_GR_VMOTB |
| 560 | 230 | DMC_FW_GR_IMOTA |
| 567 | 237 | DMC_FW_GR_IMOTB |
| 555 | 22B | DMC_FWSPEC_POS_A |
| 559 | 23F | DMC_FWSPEC_POS_B |
| 65535 | FFFF | END_OF_HK_LIST_ID |

Switch-on the spectro filter wheel controller (*DMC_SWON_FW_SPEC*).
Make a complete turn (*DMC_MOVE_SPEC_FW_STEP* with parameter = 0x600)
Stop the housekeeping acquisition.

To acquire all housekeeping related to the photo filter wheel, start a diagnostic housekeeping acquisition with the following measures:

| HK ID (dec) | HK ID (hex) | HK ID |
|---|---|---|
| 211 | D3 | DMC_FW_PHOT_CTRL |
| 256 | 100 | DMC_FWGRAT_HALL_A |
| 257 | 101 | DMC_FWGRAT_HALL_B |
| 556 | 22C | DMC_FW_GR_VMOTA |
| 564 | 234 | DMC_FW_GR_VMOTB |
| 560 | 230 | DMC_FW_GR_IMOTA |
| 567 | 237 | DMC_FW_GR_IMOTB |
| 563 | 233 | DMC_FWPHOT_POS_A |
| 569 | 239 | DMC_FWPHOT_POS_B |
| 65535 | FFFF | END_OF_HK_LIST_ID |

Switch-on the spectro filter wheel controller (*DMC_SWON_FW_PHOTO*).
Make a complete turn (*DMC_MOVE_PHOTO_FW_STEP* with parameter = 0x600)
Stop the housekeeping acquisition.

#### 4.4.11.4 Summary of commands and telemetry

**Trigger Commands:**

- 39 DMC_SWOFF_GRAT_CONT
- 58 DMC_SWON_FW_SPEC
- 64 DMC_MOVE_SPEC_FW_LOC
- 65 DMC_MOVE_SPEC_FW_STEP

**Write Commands:**

- 145 DMC_WRT_FW_SPEC_CONF_PAR

**HK nominal:**

- 210 DMC_FW_SPEC_CTRL
- 255 DMC_FWSP_CUR_POS
- 256 DMC_FWGRAT_HALLA
- 257 DMC_FWGRAT_HALLB

**HK diag:**

- 555 DMC_FW_SPEC_POSA

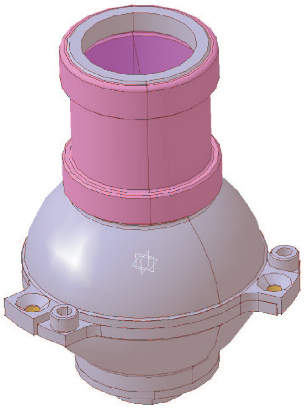
- 559 DMC_FW_SPEC_POSB
- 563 DMC_FW_PHOT_POSA
- 559 DMC_FW_PHOT_POSB
- 556 DMC_FW_GR_VMOTA
- 560 DMC_FW_GR_IMOTA
- 564 DMC_FW_GR_VMOTB
- 567 DMC_FW_GR_IMOTB

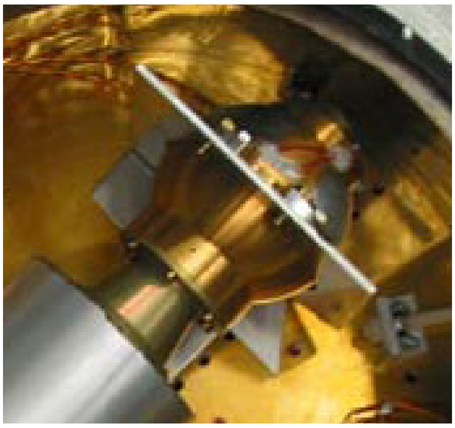
### 4.4.12 The calibration sources : general description

The calibration source is a small sphere with high reflectivity coating at the center of which is placed the light source element. A drawing and a picture of the calibration source is shown hereunder. The light source is actually a platinum resistance (PT500) hold at the center of the sphere by small isolating feet and a set of Kevlar wires. Hence, the heat losses from the heater to the environment is minimised as well as the required energy for heating. However, this also induces that the time needed for cooling down the heater is more important.

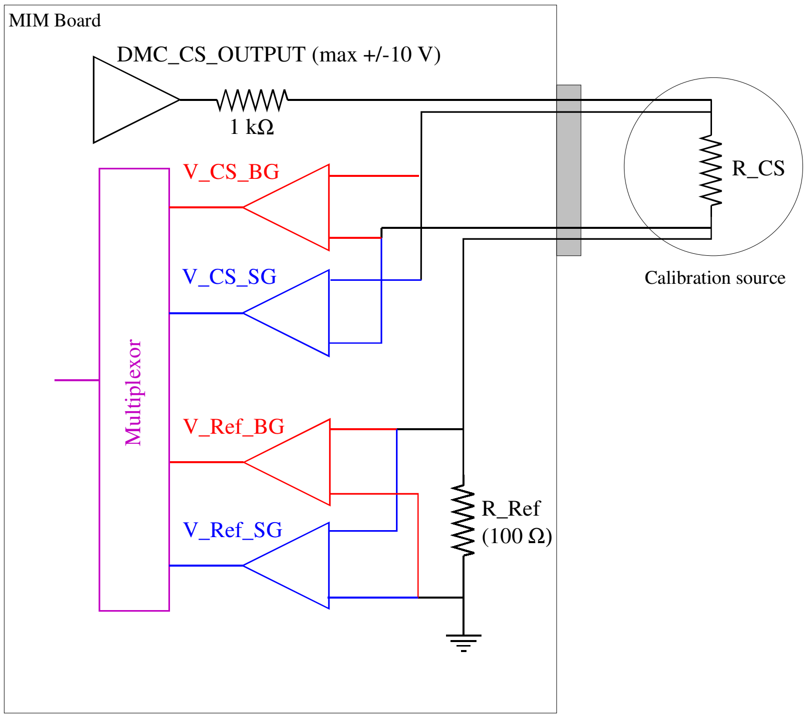
The resistance measurement of the heater is also used to determine its temperature. That is why the heater is connected in 4-wires configuration. A calibration curve is required to translate the resistance measurement to temperature.

There are two heaters, one nominal and one redundant, each connected to nominal and redundant MIM boards respectively.





The calibration sources must be controlled in order to ensure high temperature stability for long time periods. Therefore, for control and verification, the temperature readings must be done also with high resolution (~1 mK). For that purpose, a particular control electronics has been implemented in the DMC, as shown in the picture below.




The controller commands a voltage amplifier having a maximum output of +/-10 V. As the maximum allowed current through the calibration source is 10 mA, a resistance of 1 kΩ is placed in the line, ensuring that the current will never be higher than 10 mA whatever the resistance of the calibration source. A 100 Ω reference resistance (R_Ref) is placed serially with the calibration source in the DMC electronics to measure the current flowing through the source. As the resistance of the source vary from about 1 to 120 Ω (depending on the temperature), the maximum current is limited to a value going from 8.2 to 9.1 mA.

The stability and repeatability in time of the calibration source measurement is important. Therefore, the excitation of the source is made using an alternative (square wave) signal and voltage measurements are made by differential measurement between positive and negative values. This allows being independent of any offset and derivation of the power and readout amplifiers. There are two modes of excitation of the source, depending if we are in a heating phase of the source or if only readout of the temperature is required (without heating power). See the figure below to illustrate the following explanations.

The excitation is an alternative square wave with a frequency of 1 Hz. In heating mode of the source, the excitation is applied continuously while its amplitude is updated at each controller step. Readouts of the calibration source temperature are made at a frequency of 0.05 Hz, which is therefore also the frequency of the controller (a controller step is done after each readout). In reading only mode, the power through the source must be reduced to a minimum and therefore, the source is only excited when a readout is required. Moreover, the amplitude of the excitation is small (100 mV ⇒ current through the source < 100 µA). The reading mode is selected when the output of the controller is lower than 327 (= 32767/100 ⇒ 100 mV) and a fortiori when the output is negative, i.e. when cooling of the source is required. This mode is also used when the controller is disabled. This threshold to switch from heating to reading only mode can also be modified by changing the value of Output Threshold parameter of the calibration source controller (see §4.2.1.5).

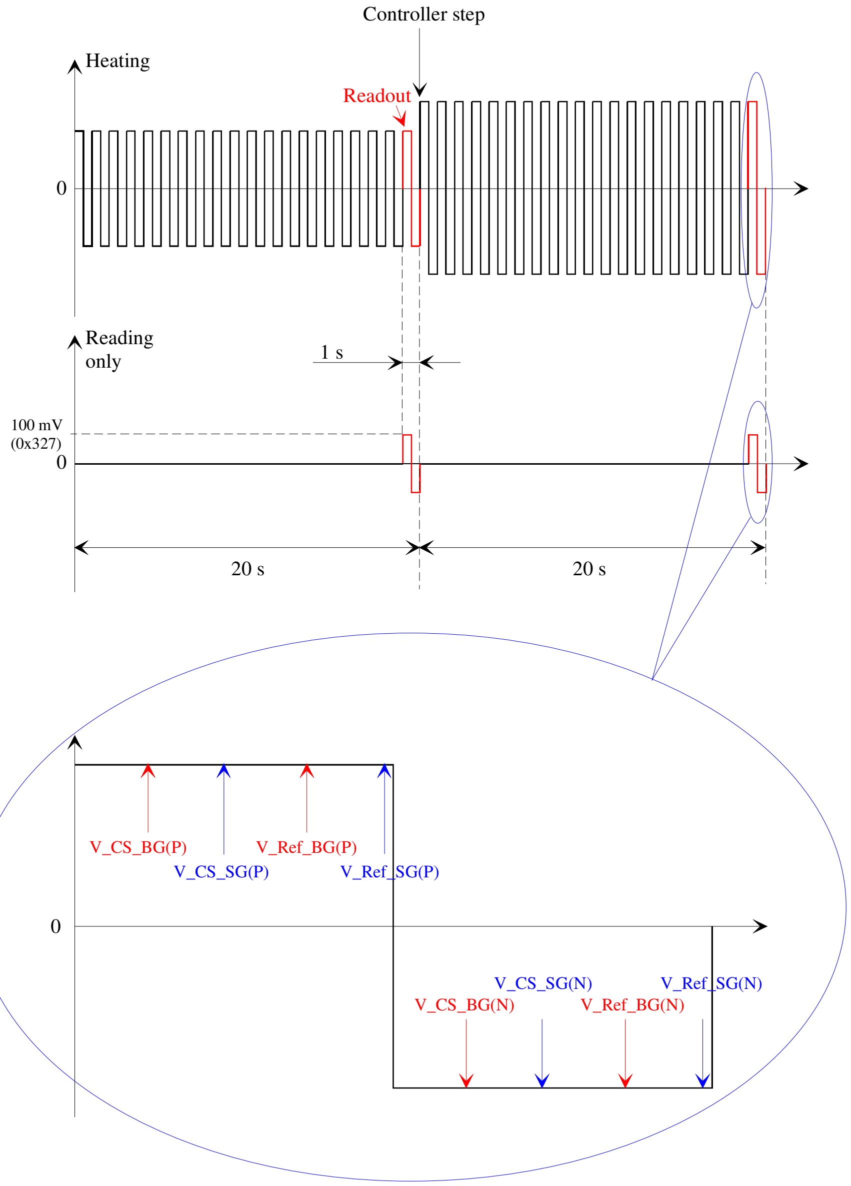
Determining the calibration source temperature is done by acquiring 4 values during the positive and negative parts of the square excitation. This is represented in the lower part of the figure. Actually, 8 values are acquired but only the represented values are required for the temperature measurement, the other being for housekeeping only. Therefore, only the 4 interesting data are represented for clarity. Refer also to the figure above to see where these data are measured.

The voltages through the source and through the reference resistance are measured to determine the calibration source resistance. In order to have high resolution for large voltage values (mainly in heating mode and at high temperature) as well as for small voltage values (reading mode and at low temperature), each voltage is measured through two amplifiers. These two amplifiers have different gain factors and are named "big gain" (BG) and "small gain" (SG). The full scale conversions for the two amplifier gains are as following :

BG → ± 32767 = ± 0.025 V

SG → ± 32767 = ± 2.5 V

With a total of 8 measured data, the calculation of the calibration source resistance is done as following. The 4 voltages (V_CS_BG, V_CS_SG, V_Ref_BG, V_Ref_SG) are obtained by subtraction of the positive and negative measurements, resulting in positive numbers in the range 0-65535. For example, V_CS_BG = V_CS_BG(P) – V_CS_BG(N).




Then a selection between the two amplification gains is done automatically to benefit from the highest resolution. The criteria are as following :

*Switching from BG to SG if*

(V_CS_BG > 64000) *OR* (V_Ref_BG > 64000) ⇒ V_CS = V_CS_SG *AND* V_Ref = V_Ref_SG

*Switching from SG to BG if*

(V_CS_SG < 620) *AND* (V_Ref_SG < 620) ⇒ V_CS = V_CS_BG *AND* V_Ref = V_Ref_BG

At switch on, the BG is selected by default.

Finally, the resistance of the calibration source is :
R_CS = (V_CS/V_Ref)*R_Ref

For clarity of the above explanation and schematics, explicit simple names have been used for all the data. However, all these data can be found in the DMC housekeeping (either nominal or diagnostic) with the following labels :

| | | | |
|---|---|---|---|
| V_CS_BG | → | DMC_CS1_VOLT_BG / DMC_CS2_VOLT_BG | (diag) |
| V_CS_SG | → | DMC_CS1_VOLT_SG / DMC_CS2_VOLT_SG | (diag) |
| V_Ref_BG | → | DMC_CS1_CUR_BG / DMC_CS2_CUR_BG | (diag) |
| V_Ref_SG | → | DMC_CS1_CUR_SG / DMC_CS2_CUR_SG | (diag) |
| R_CS | → | DMC_CS1_RES_VALUE / DMC_CS2_RES_VALUE | (nom) |
| Controller output | → | DMC_CS1_OUTPUT / DMC_CS2_OUTPUT | (nom) |

All data are 16 bits integers with conversion given above and in §5.3 excepted the resistance of the calibration source (DMC_CS1_RES_VALUE / DMC_CS2_RES_VALUE) for which the housekeeping value is directly the resistance measurement with 1 digit = 0.1 mΩ.

### 4.4.13 Using the calibration sources

The controller is switched on by sending the command *DMC_SWON_BB_1(2)_CONT*, resulting in starting the calibration source resistance measurement in reading mode only. The resistance is then updated in the housekeeping each time a readout is done, i.e. every 20 s. It is switched off using the command *DMC_SWOF_BB_1(2)_CONT*.

#### 4.4.13.1 Nominal mode operation

The controller can be enabled using the following command *DMC_ENABLE_BB_1(2)_CONT* and disabled using *DMC_DISABLE_BB_1(2)_CONT*. The source is then commanded by specifying a resistance value (conversion from T to R must be done by the user) with the command *DMC_SET_TEMP_BB_1(2)*. Commanding units are the same than reading units, i.e. 1 digit = 0.1 mOhm. Then switching to heating mode occurs if required and the amplitude of the output is calculated through a PI regulator (no derivative term) with limited integral accumulator (see §4.2.1.5 for details on the controller parameters).

#### 4.4.13.2 Open loop operation

The calibration source can be also operated in open loop, i.e. by commanding a specified heating power through the source, with no regulation anymore. However, the square wave output strategy is still used but with the commanded amplitude. Commanding an output of 2V (= 6554 digits) witch results in a square power output of ±2V (±6554 digits)
Open loop output power can be commanded using the command *DMC_SET_BB_1(2)_VOLTAGE*.

#### 4.4.13.3 Summary of commands and telemetry

**Trigger Commands:**

- 68 DMC_SWON_BB1_CONT
- 69 DMC_SWOFF_BB1_CONT
- 91 DMC_ENABLE_BB1_CONT
- 92 DMC_DISABLE_BB1_CONT
- 70 DMC_SET_BB1_TEMP
- 71 DMC_SET_BB1_VOLTAGE
- 72 DMC_SWON_BB2_CONT
- 73 DMC_SWOFF_BB2_CONT
- 93 DMC_ENABLE_BB2_CONT
- 94 DMC_DISABLE_BB2_CONT
- 74 DMC_SET_BB2_TEMP
- 75 DMC_SET_BB2_VOLTAGE

**Write Commands:**

- 147 DMC_WRT_CS1_CONF_PAR
- 148 DMC_WRT_CS2_CONF_PAR

**HK nominal:**

- 213 DMC_CS1_CTRL_STA
- 459 DMC_CS1_TARGET
- 445 DMC_CS1_RES_VALUE
- 446 DMC_CS1_OUTPUT
- 214 DMC_CS2_CTRL_STA
- 460 DMC_CS2_TARGET
- 447 DMC_CS2_RES_VALUE
- 448 DMC_CS2_OUTPUT

**HK diag:**

- **Reference voltages**
- 522 DMC_CS1_VOLT_0V
- 523 DMC_CS1_VOLT_N5V
- 524 DMC_CS1_VOLT_P5V
- 538 DMC_CS2_VOLT_0V
- **Measures**
- 525 DMC_CS1_VOLT_DAC_OUT
- 526 DMC_CS1_VOLT_SG
- 527 DMC_CS1_VOLT_BG

- 528 DMC_CS1_CUR_SG
- 529 DMC_CS1_CUR_BG
- 541 DMC_CS2_VOLT_DAC_OUT
- 542 DMC_CS2_VOLT_SG
- 543 DMC_CS2_VOLT_BG
- 544 DMC_CS2_CUR_SG
- 545 DMC_CS2_CUR_BG

### 4.4.14 Using the FPU temperature sensors

There are 7 FPU temperature sensors:

- One on the chopper
- One on the grating
- One on the spectro FW
- One on the photo FW
- Two in the FPU
- One on the calibration source housing

To avoid power dissipation, the temperature sensors are not measured at switch-on of DMC. You must send the command *DMC_SWON_TEMP_SENSORS* to start the measure.
Each measure cycle takes around 1 minute (the hk values are updated every minute but are sent in every nominal hk packet).
All the measures are not updated at the same time in the cycle.

There are two amplifier circuits that can be used to make the measure depending on the current resistor value of the sensor. They are called 'high gain' and 'low gain'. In DMC_FPU_T_SENS_ST, you can find out which gain has been used for the measure. No matter the gain that has been used, the temperature sensor measure will always be expressed in the same units (ohms) but the error of the measure will be different.

The resistor measure made by DMC is not calibrated (there is an error of around 1%). Two calibrations must be done on ground (one for each of the gain) in order to provide a correction table. Note that once the FPU is cold, all measures will be done with the low gain and thus, only one correction table is mandatory to operate the instrument in cold.

Furthermore, DMC has two different circuits to make the measures, the first one is measuring:

- DMC_CHOPPER_TEMP
- DMC_CAL_SRC_TEMP
- DMC_FPU_T1_TEMP

- DMC_FPU_T2_TEMP

The second one is measuring:

- DMC_GRATING_TEMP
- DMC_FW_SPEC_TEMP
- DMC_FW_PHOT_TEMP

Each of the circuit must have its own correction table.

#### 4.4.14.1 Summary of commands and telemetry

**Trigger Commands:**

- 95 DMC_SWON_TEMP_SENSORS
- 96 DMC_SWOF_TEMP_SENSORS

**Write Commands:**

- none

**HK nominal:**

- 404 DMC_FPU_T_SENS_ST
- 405 DMC_FW_SPEC_TEMP
- 406 DMC_FW_PHOT_TEMP
- 407 DMC_CHOPPER_TEMP
- 408 DMC_GRATING_TEMP
- 426 DMC_FPU_T1_T
- 427 DMC_FPU_T2_T
- 429 DMC_CAL_SRC_TEMP

**HK diag:**

- 571 DMC_T_SE_SRC1_LG
- 572 DMC_T_SE_SRC1_HG
- 573 DMC_T_SE_SRC1_V1
- 574 DMC_T_SE_SRC1_V2
- 575 DMC_T_SE_SRC2_LG
- 576 DMC_T_SE_SRC2_HG
- 577 DMC_T_SE_SRC2_V1
- 578 DMC_T_SE_SRC2_V2
- 619 DMC_TS_FW_SPEC_V
- 620 DMC_TS_FW_PHOT_V
- 621 DMC_TS_GRAT_V
- 622 DMC_TS_CHOP_V
- 623 DMC_TS_FPU_T1_V

- 624 DMC_TS_FPU_T2_V
- 625 DMC_TS_BB_V

### 4.4.15 Using the CRE temperature sensors

As soon as a DEC is switched on, all the temperature sensors connected to it are operated. All the computation is performed in the DEC BASE FPGA. The operation is the same as for the FPU temperature sensors.

#### 4.4.15.1 Summary of commands and telemetry

**Trigger Commands:**

- none

**Write Commands:**

- none

**HK nominal:**

- 288 DMC_DECB_TS_ST_3
- 295 DMC_DECB_TS_1_3
- 296 DMC_DECB_TS_2_3
- 322 DMC_DECB_TS_ST_4
- 329 DMC_DECB_TS_1_4
- 330 DMC_DECB_TS_2_4
- 356 DMC_DECR_TS_ST_1
- 363 DMC_DECR_TS_1_1
- 364 DMC_DECR_TS_2_1
- 390 DMC_DECR_TS_ST_2
- 397 DMC_DECR_TS_1_2
- 398 DMC_DECR_TS_2_2

**HK diag:**

- 579 DMC_DB_TS12CBS_3
- 580 DMC_DB_TS12CSS_3
- 581 DMC_DECB_TS1_V_3
- 582 DMC_DECB_TS2_V_3
- 589 DMC_DB_TS12CBS_4
- 590 DMC_DB_TS12CSS_4
- 591 DMC_DECB_TS1_V_4
- 592 DMC_DECB_TS2_V_4
- 599 DMC_DR_TS12CBS_1

- 600 DMC_DR_TS12CSS_1
- 601 DMC_DECR_TS1_V_1
- 602 DMC_DECR_TS2_V_1
- 609 DMC_DR_TS12CBS_2
- 610 DMC_DR_TS12CSS_2
- 611 DMC_DECR_TS1_V_2
- 622 DMC_DECR_TS2_V_2

### 4.4.16 Warm electronic temperature sensors

There are 4 temperature sensors in the warm electronic area:

- 1 on each DEC DC/DC
- 1 on the DMC DC/DC
- 1 on the CPU board

They are measured all the time as long as DMC or DECs are powered on. Their value is given by a voltage and must be converted to degrees according to TBD conversion formula.

#### 4.4.16.1 Summary of commands and telemetry

**Trigger Commands:**

- none

**Write Commands:**

- none

**HK nominal:**

- 284 DMC_DECB_DCDC_T3
- 352 DMC_DECR_DCDC_T1
- 413 DMC_DCDC_TEMP
- 414 DMC_DSP_TEMP

**HK diag:**

- none

### 4.4.17 SPU housekeeping

There are 7 sensors located in the SPU that are connected to DMC. They are measured all the time as long as DMC and SPU are powered on.

#### 4.4.17.1 Summary of commands and telemetry

**Trigger Commands:**

- none

**Write Commands:**

- none

**HK nominal:**

- 419 DMC_SPU_PSU_15V
- 420 DMC_SPU_SWL_TEMP
- 421 DMC_SPU_LWL_TEMP
- 422 DMC_SPU_PS_TEMP
- 423 DMC_SPU_VCC_CUR
- 424 DMC_SPU_VCC_VOL
- 425 DMC_SPU_VP_CUR

**HK diag:**

- none

### 4.4.18 Using diagnostic housekeeping

To start using the diagnostic housekeeping, one should:

1. Upload the list of measures to be monitored (*DMC_WRITE_DIAG_HK_LIST*). Make sure that the last value of the list is END_OF_HK_LIST_ID (0xFFFF).
2. Start the diagnostic housekeeping (*DMC_START_DIAG_HK*)
3. When done, stop the diagnostic housekeeping (*DMC_STOP_DIAG_HK*)

#### 4.4.18.1 Summary of commands and telemetry

**Trigger Commands:**

- 76 DMC_START_DIAG_HK
- 77 DMC_STOP_DIAG_HK

**Write Commands:**

- 141 DMC_WRT_DIAG_HK_LIST

**HK nominal:**

- 462 DMC_HK_DIAG_STAT
- 463 DMC_HK_DIAG_PERI

**HK diag:**

- none

### 4.4.19 Using the sequencer

#### 4.4.19.1 Selecting the synchronization source

The synchronization source selection has two effects:

- It determines the synchronization signal for the execution of the sequence commands
- It determines the instant where the mechanisms start moving.

The DMC_SYNCHRONIZE_ON_DET trigger commands allows you to select the synchronization source. It has 3 nominal values:

- Blue DEC: the sequencer WAIT statements are waiting for the end of ramps from Blue DEC (the sequencer waits for the reception of the penultimate 1355 packet of the ramp). The mechanisms start moving when the destructive readout synchronization signal is received (by default, we take the signal coming from the Blue DEC group 3)
- Red DEC: the sequencer WAIT statements are waiting for the end of ramps from Red DEC (the sequencer waits for the reception of the penultimate 1355 packet of the ramp). The mechanisms start moving when the destructive readout synchronization signal is received (by default, we take the signal coming from the Red DEC group 1)
- BOLC: the sequencer WAIT statements are waiting a readout from BOLC (the sequencer waits for the reception of the 1355 packet whose blockNum = 0). The mechanisms start moving when the destructive readout synchronization signal is received

When the DMC_SYNCHRONIZE_ON_DET is received, it modifies these 2 values:

- DMC_SEQ_OPTIONS
- The word 0 of DMC_WRT_TIMING_FPGA_PAR: sync_src_sel_reg (note that this value is no more modifiable through the write command but only through the trigger command).

At power-up, the sequencer is synchronized on the blue DEC and the mechanisms use the internal synchronization source (256Hz signal generated in the timing FPGA).

#### 4.4.19.2 Executing a sequence

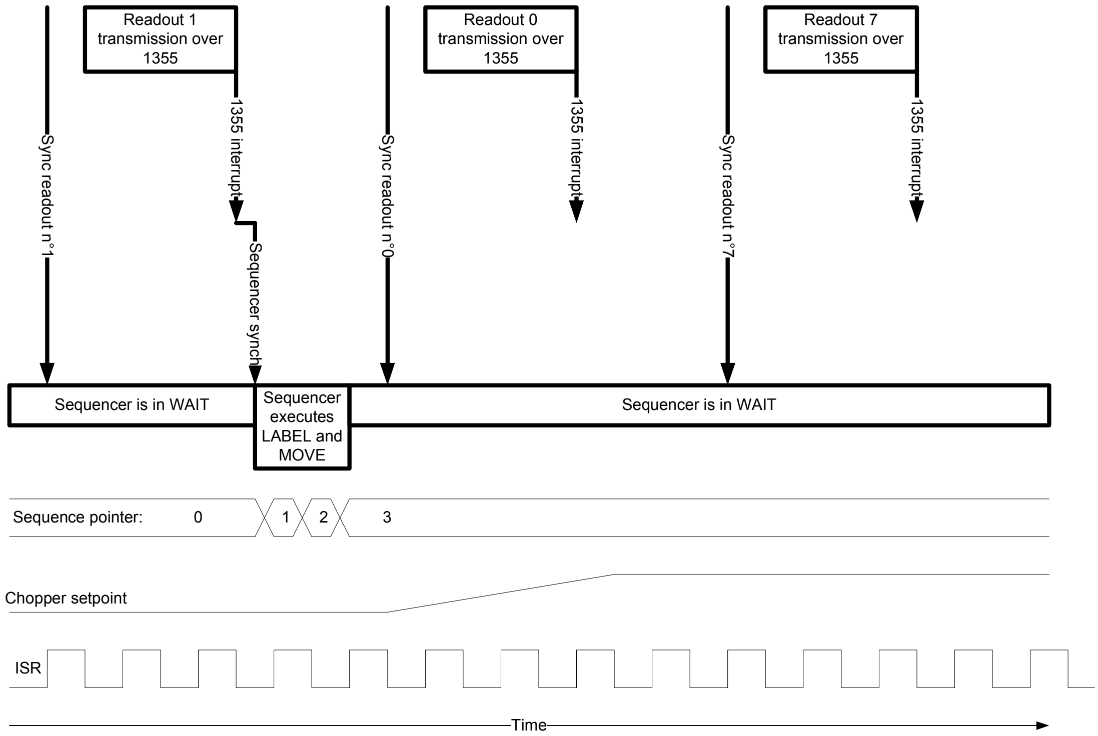
To execute a sequence, you should:

- Have at least one synchronization source switched-on and sending data
- Send a DMC_SYNCHRONIZE_ON_DET command
- Upload a sequence

Once the execution has started, the Sequencer executes all the command one after the other (usually, the execution of such commands takes only a few micro-seconds). The DMC_SEQ_POINTER shows which command the sequencer is currently executing.

The DMC_WAIT command is used to synchronize the Sequencer with the science data (and thus, synchronize the movement of a mechanism with the science data).

The drawing below shows how the synchronization works in the spectroscopy mode. Note that, in these diagrams, the number of ISR, the time needed for command execution, … are not representative. The diagram just shows the sequencing of events.




Everytime the DMC receives the penultimate readout of a ramp, the Sequencer receives a 'Sequence synchro'. At that time, the Sequencer checks wether it has to wait for another ramp or if it can execute the next commands. In the second case, the Sequencer executes all the commands until it reaches another DMC_WAIT statement.

In our example, the Sequencer has been given the following sequence:

DMC_WAIT(1)

DMC_LABEL(2)

DMC_MOVE_CHOPPER_REL(1000)

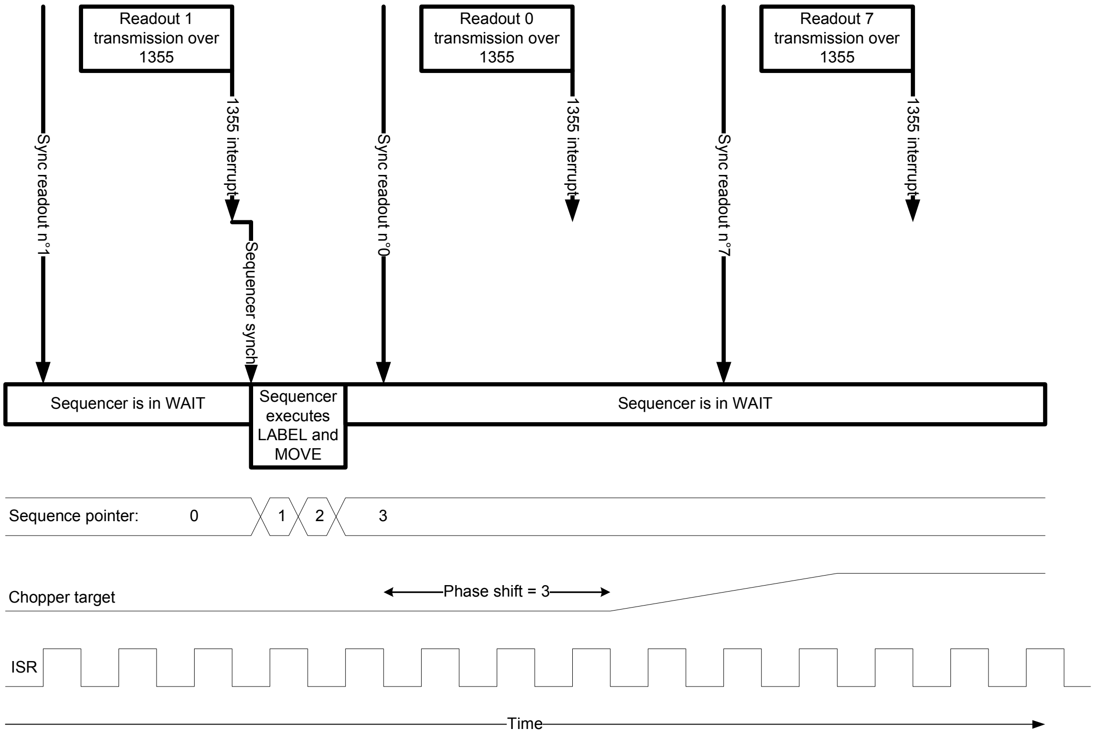
DMC_WAIT(1)

During the first ramp, the Sequence pointer is 0 (the Sequencer is waiting for the first ramp). Once the readout n°1 of the ramp is received, the Sequencer executes the DMC_LABEL command (it is only a few instructions so the Sequence pointer will be 1 for only a few micro-seconds) and then, the DMC_MOVE_CHOPPER.

The DMC_MOVE_CHOPPER command is only programming the ISR to start a move of the chopper at the next synchro signal. This is also only a few commands so, very quickly, the sequencer moves to the position 3 where it waits for the next ramp.

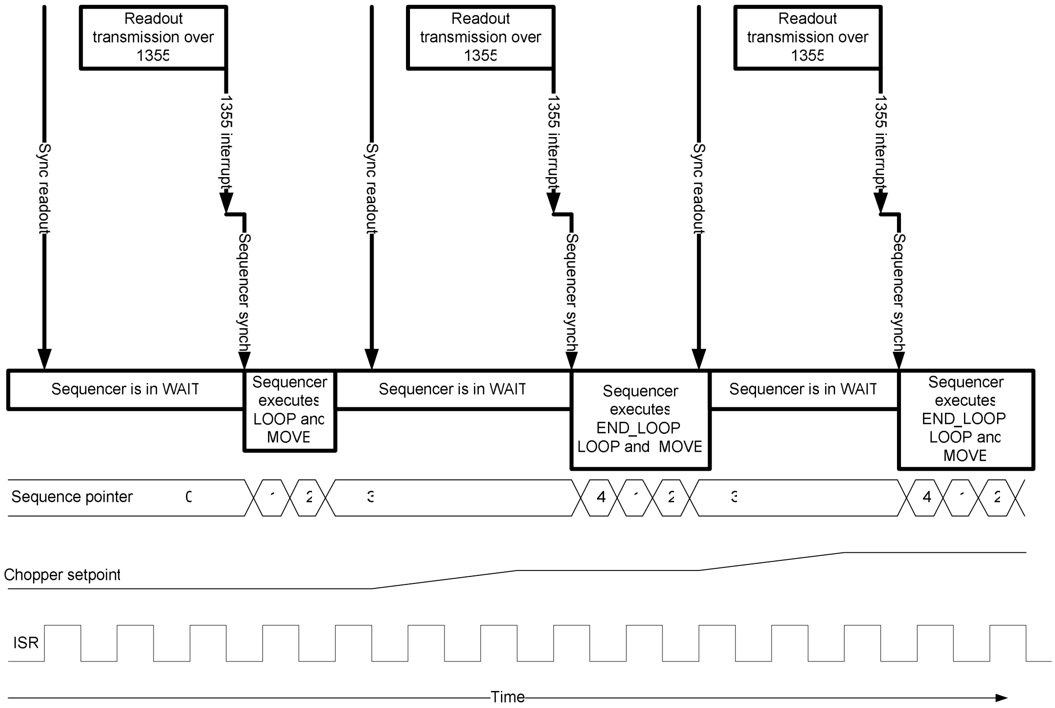
The ISR is executing at 8KHz and is always checking if a synchro has been received. As soon as it gets one, it will start to modify the setpoint of the chopper (and thus, the chopper will also start moving).

In the drawing below, we show the same process in the case a phase_shift has been set to 3 (by setting the phase_shift_reg to 2!) (Phase_shift is one of the parameters of DMC_WRT_TIMING_FPGA_PAR). The phase_shift is the delay (expressed in number of ISR execution) between the reception of the synchronization signal from the detectors and the start of the movement of the mechanism.




The diagram below shows the execution of a sequence in photometry mode (with phase shift = 0). The Sequencer has been given the following sequence:

DMC_WAIT(1)
DMC_LOOP(4)
DMC_MOVE_CHOPPER_REL(500)
DMC_WAIT(1)
DMC_END_LOOP()




### 4.4.19.3 Summary of commands and telemetry

**Trigger Commands:**

- 0 DMC_LOOP
- 1 DMC_END_LOOP
- 2 DMC_WAIT
- 3 DMC_END_SEQUENCE
- 4 DMC_LABEL
- 5 DMC_START_SEQUENCE
- 6 DMC_ABORT_SEQUENCE
- 10 DMC_SYNCHRONIZE_ON_DET

**Write Commands:**

- 129 DMC_WRT_SEQ_BUFFER
- 130-139 DMC_WRT_SEQ_BUFFER_0-9

**HK nominal:**

- 197 DMC_SEQ_STATUS
- 215 DMC_SEQ_OPTIONS
- 216 DMC_SEQ_POINTER
- 217 DMC_SEQ_LOOP_ID0

- 218 DMC_SEQ_LOOP_ID1
- 219 DMC_SEQ_LOOP_ID2
- 220 DMC_SEQ_LOOP_ID3
- 221 DMC_SEQ_LOOP_ID4
- 222 DMC_SEQ_WAIT_IND
- 223 DMC_SEQ_LABEL

**HK diag:**

- none

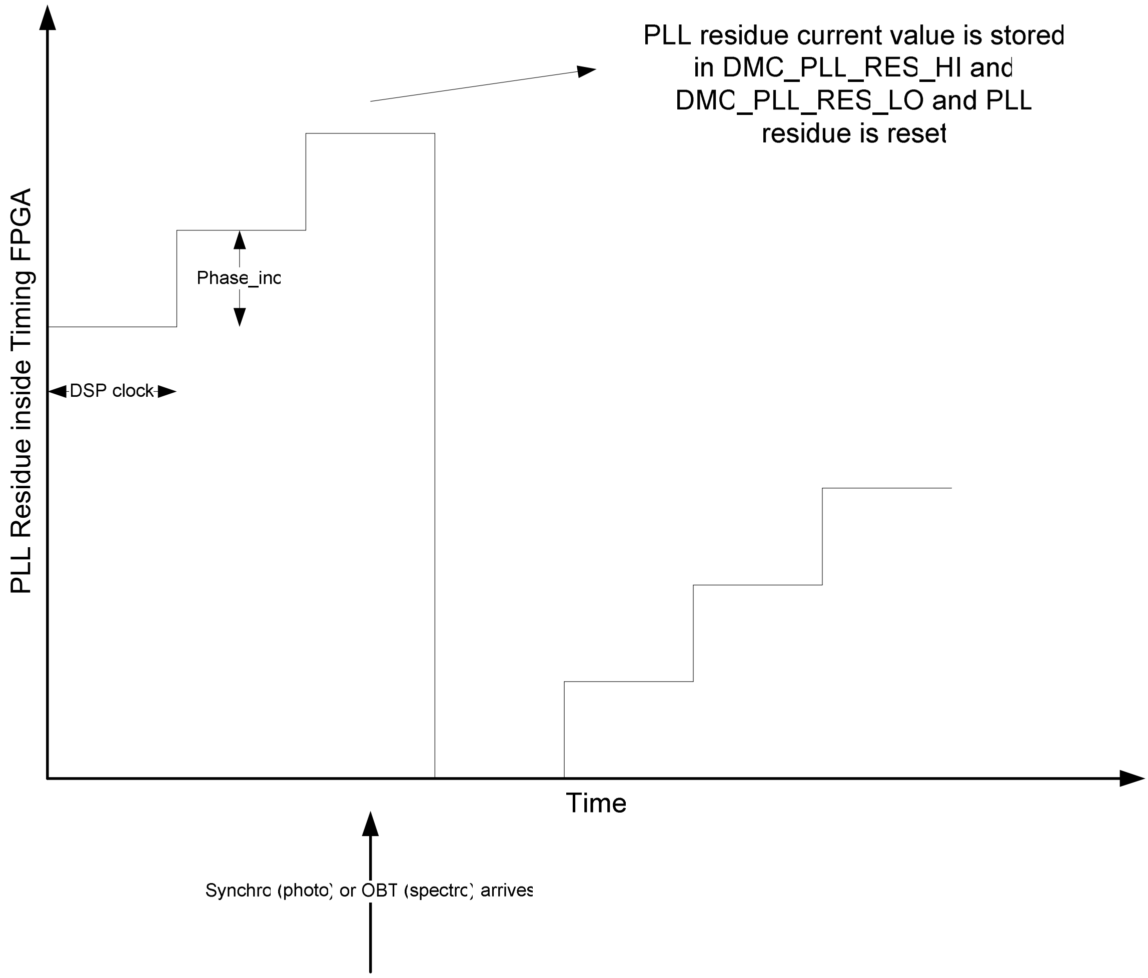
### 4.4.20 Adjusting the timing parameters

At regular interval during the mission, the phase_inc parameter shall be adjusted to take into account the frequency drift of OBT and/or DSP clock. This is mandatory to ensure that the mechanisms move in phase with the readouts.

#### 4.4.20.1 Adjusting phase_inc in photometry

- Switch-on BOLC and set the photometry timing mode (keep the default 40Hz readout frequency for BOLC).
- Get DMC_PLL_RES_HI and DMC_PLL_RES_LO from nominal HK.
- Compute DMC_PLL_RES by appending DMC_PLL_RES_RES_HI and DMC_PLL_RES_LO to form a 48bit number
- Compute the PLL_ERROR = 0x40FFFFFE6420 – DMC_PLL_RES
- Compute correction = PLL_ERROR/450000
- Compute new phase_inc = phase_inc + correction. If DMC_PLL_RES_HI >= 0x4100, the new phas_inc shall be smaller than the old one. If DMC_PLL_RES_HI < 0x4100, the new phase_inc shall be bigger than the old one.
- Apply the new parameters using the nominal procedure

In photometry mode, the PLL internal frequency is the DSP clock (18MHz). It means that the PLL residue is incremented by phase_inc at every DSP clock. Every time a synchronization signal is received from BOLC, the current PLL residue value is stored in the DMC_PLL_RES_HI and DMC_PLL_RES_LO. You can not see the PLL residue incrementing, you can only see a snapshot of its value taken when the sync arrives.




So, if BOLC is working at a perfect 40Hz, with a phase_inc of 158818346 (nominal value), the PLL residue shall be = 18000000*158818346/40 = 0x40FFFFFE6420. This is a 48bits number that is stored in 2 HK values (DMC_PLL_RES_HI = 0x40FF and DMC_PLL_RES_LO = 0xFFFE6420). This value can be considered as the 'ideal value' for BOLC at 40 Hz.

DMC_PLL_RES_HI can be viewed as a counter counting at 665600Hz. The PLL residue can never deviate by more than a period of this counter. So, the DMC_PLL_RES_HI accepted value can only be 0x40FF or 0x4100. If the PLL residue is out of this range, you will have additional delays on the mechanisms movement but you could also observe bad behaviours in the analog housekeeping and/or in the calibration source operation.

So, as soon as the PLL deviate by more than half of a period, you should adapt the phase_inc as explained above. So:

Minimum PLL residue : 0x40FF80000000

Maximum PLL residue : 0x41007FFFFFF

If the PLL residue stays within these limits, the influence of the PLL on internal timings will never be bigger than half a period of 665600Hz (= 0,75µs)

#### 4.4.20.2 Adjusting phase_inc in spectrometry

- Set the spectrometry timing mode.
- Get DMC_PLL_RES_HI and DMC_PLL_RES_LO from nominal HK.
- Compute DMC_PLL_RES by appending DMC_PLL_RES_RES_HI and DMC_PLL_RES_LO to form a 48bit number
- Compute the PLL_ERROR = 0x4FFFFFFF7 – DMC_PLL_RES
- Compute correction = DMC_PLL_ERROR/137
- Compute new phase_inc = phase_inc + correction. If DMC_PLL_RES_HI >= 5, the new phas_inc shall be bigger than the old one. If DMC_PLL_RES_HI < 5, the new phase_inc shall be smaller than the old one.
- Apply the new parameters using the nominal procedure

In spectrometry mode, the PLL internal frequency is also the DSP clock (18MHz). It means that the PLL residue is incremented by phase_inc at every DSP clock. At every OBT clock, the current PLL residue value is stored in the DMC_PLL_RES_HI and DMC_PLL_RES_LO.

So, if OBT is at a perfect 131072Hz, with a phase_inc of 156374987 (nominal value), the PLL residue shall be = 18000000*156374987/131072 = 0x4FFFFFFF7. This is a 48bits number that is stored in 2 HK values (DMC_PLL_RES_HI = 0x4 and DMC_PLL_RES_LO = 0xFFFFFFF7). This value can be considered as the ‘ideal value’ for OBT at 131072Hz.

DMC_PLL_RES_HI can be viewed as a counter counting at 655360Hz. The PLL residue can never deviate by more than a period of this counter. So, the DMC_PLL_RES_HI accepted value can only be 0x4 or 0x5. If the PLL residue is out of this range, you will have additional delays on the mechanisms movement but you could also observe bad behaviours in the analog housekeeping and/or in the calibration source operation.

So, as soon as the PLL deviate by more than half of a period, you should adapt the phase_inc as explained above. So:

Minimum PLL residue : 0x480000000

Maximum PLL residue : 0x57FFFFFF

If the PLL residue stays within these limits, the influence of the PLL on internal timings will never be bigger than half a period of 655360Hz (= 0,76µs)

### 4.4.21 FPGA status register diagnostic mode

OBS 6.028 contains a dedicated diagnostic mode to study the sampling of the FPGA status register.

In this mode, the ISR is replaced by a completely different piece of code that is sampling the FPGA status register 5 times along a period of 60 µs.

When outside of this mode, v6.028 has full flight capability.

While in the diagnostic mode, the nominal ISR is not executed anymore. This means that:

- Chopper, grating, FW controllers are not executed anymore
- Analog HK, temperature sensors and calibration sources are not working anymore
- PLL residue is not updated
- DMC_SYNC_COUNT and CRDCCP are updated 5 times faster

So, it is mandatory to disable all controllers before using the diagnostic mode. Controllers should be enabled again only when you have exited the diagnostic mode.

The diagnostic mode concentrates on the sampling of bit 8 and 9 of the FPGA status register:

Each measure of Bit8 and 9 of the FPGA status register will be recorded (this means that we have an non-steady sampling frequency of 40KHz).

The recorded values of these bits will be stored in the photometer science data (blockId = 1) as follows:

Each 32bits word of science data will contain: "0000 0000 000a bcde 0000 0000 000f ghij" where:

a is the first sampling of bit 8 in the ISR
b is the second .........................
c
d
e is the fifth
f is the first sampling of bit 9 in the ISR
g is the second .....
h
i
j

First pixel contains the sampling performed at the first interrupt following the reception of the first packet of a readout

Second pixel .... at the second interrupt ....

Since there are 208 ISR between two readouts, there should be 208 pixels used in each packet.

Note that there might be a jitter in the position of pixels in the packet but we will do our best to minimize it (probably between 0 and 1 position). However, what we want to observe in this test is the value of the register varying during the ISR execution; this would not be affected by this jitter.

The data should be all zeroes except for:
- 1 pixel where the 10 bits shall be set to 1 if phase_shift == 0
- 2 pixels where 5 bits shall be set to 1 if phase_shift > 0
- any other value should trigger our attention. That should be quite easy for you to detect.

BOLC must be configured to send science data. The content of blockID = 1 will be replaced by the samples.

# 5 Housekeeping

Note : Housekeeping acquisition are not performed inside critical section and are not protected by any other synchronisation mechanism. That means that, any task may be modifying a variable while the housekeeping task is copying it into the Hk Buffer. So, a few inconsistencies may appear in the housekeeping measures. They shall be very seldom.

Protection against these inconsistencies is not recommended since it would affect the real-time behaviour of the onboard software.

## 5.1 Offset and gain correction of analog housekeeping

Many of the housekeeping measure represents analog measures perform by ADC. The offset and gain errors of the ADC can be corrected by comparing the measure with a “zero volt” and a “reference” channel.

To have the best measure, the “zero volt” channel value should be subtracted from each measurement from the corresponding ADC such that the measure of 0V really displays 0V. Once this is done, if a measurement from a trusted fixed voltage reference is available, its value can be used to compensate each channel reading.

True value = (readout value – zero volt reference readout value) * theoretical reference / reference readout value

| Subsystem and particular HK IDs | Zero volt reference channel to use | Fixed Reference channel to use | Theoretical reference |
|---|---|---|---|
| FPU_TEMP (V and I measures only)<br>HK<br>SPU_HK<br>GRAT (except HK ID 539 | DMC_REF_VOLT_0V | DMC_REF_VOLT_5V | 5V |

| | | | |
|---|---|---|---|
| and 540)<br>CHOP<br>FW | | | |
| CS + HK ID 539 and 540 | DMC_CS1_VOLT_0V | DMC_CS1_VOLT_N5V<br>DMC_CS1_VOLT_P5V | -4V<br>+4V |
| DEC group 1 (HK ID 333 – 347) | DMC_DECR_V0V_1 | None | |
| DEC group 1 (HK ID 348 – 354, 599 - 608) | DMC_DECR_REF_0V1 | DMC_DECR_R5V_1 | +5V |
| DEC group 2 (HK ID 367 – 381) | DMC_DECR_V0V_2 | None | |
| DEC group 2 (HK ID 382 – 388, 609 - 618) | DMC_DECR_REF_0V2 | DMC_DECR_R5V_2 | +5V |
| DEC group 3 (HK ID 265 - 280) | DMC_DECB_V0V_3 | None | |
| DEC group 3 (HK ID 281 – 287, 579 - 588) | DMC_DECB_REF_0V3 | DMC_DECB_R5V_3 | +5V |
| DEC group 4 (HK ID 299 – 314) | DMC_DECB_V0V_4 | None | |
| DEC group 4 (HK ID 315 – 321, 589 - 598) | DMC_DECB_REF_0V4 | DMC_DECB_R5V_4 | +5V |

In example to get an accurate value of DMC_FW_GR_IMOTA, you should compute:
IMOTA = (DMC_FW_GR_IMOTA – DMC_REF_VOLT_0V)*5/DMC_REF_VOLT_5V

Or, to get DMC_DECR_V0BIAS1, you should compute:
V0BIAS = (DMC_ DECR_V0BIAS1 – DMC_DECR_V0V_1)

**Important note:**

- These correction can be applied only on voltage and current measures
- You should correct the hk measure only if you want the best accuracy. In most of the case, it is not necessary
- For FPU temperature sensors and the CS resistor value, the offset error is already cancelled since we take a negative and a positive measure.

## 5.2 Internal sampling frequency of housekeeping values

### 5.2.1 MEC analog housekeeping values

All of them are sampled at 128Hz except the ones that are used in the mechanisms controller interrupt routine. These values are sampled at 8KHz:

- DMC_CHOP_CUR_POS,
- DMC_GRAT_CUR_POS,

- DMC_FWGRAT_HALLA,
- DMC_FWGRAT_HALLB

## 5.3 List of available measures

In the following table, we present the set of measures that are implemented in the current version of the onboard software.

The IDs lower than 512 are included in the Nominal HK packet. The IDs higher or equal to 512 are available in Diagnostic HK only.

A special ID is used to mark the end of a diagnostic HK list : END_OF_HK_LIST_ID = 0xFFFF.

### 5.3.1 How to use the list of HK measures

In the next section, the list of HK measures is presented. This section tells you how to use this information.

| 464 | DMC_LAST_ERR_ID | OBSW |
|---|---|---|
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | Nominal HK Only |
| | Useful size (in bytes): | 1 (4bits) |
| Description: Last Error Buffer Index : Indicates the current position in the Last Errors Buffer. Index is 0 based and indicates the next position to be filled. (Note: index 0 is DMC_LAST_ER_BF1) | | |

- 464 is the numerical identifier of the measure and is also its position in the nominal HK packet that is sent every 2 seconds to DPU.
- DMC_LAST_ERR_ID is the alphanumerical identifier of the measure. This is the name that will appear in SCOS2000 display.
- OBSW is identifying the subsystem to which this measure is related. The list of subsystem is:
  - BOLC: the HK coming from BOLC and the status words of the tasks related to BOLC
  - CHOP: chopper
  - COM: communication with DPU and SPU
  - CS: calibration sources

  - DEC: the HK coming from DEC and the status words of the tasks related to DEC
  - FPU_TEMP: FPU temperature sensors
  - FW: filter wheels
  - GRAT: grating
  - GRAT_FW: some measures are common to grating and filter wheels
  - HK: various analog housekeeping
  - OBSW: On board software status variables
  - SEQ: Sequencer
  - SPARE: all spare measures
  - SPU_HK: analog housekeeping from SPU
  - TIME: timing and time-stamping
- SCOS2000 display: how the value should be displayed and converted in SCOS2000
- Validity at start-up:
  - Valid: the value is meaningful as soon as DMC is switched on
  - Invalid: the value is meaningful only after some operation has been performed (switch-on of a device …). Since it is invalid, the measure will contain 0xFFFFFFFF
- Validity during execution: gives you the operation to perform to make this measure meaningful.
- Available in:
  - Nominal HK only: This measure is only available in the nominal housekeeping. It can not be included in a diagnostic list.
  - Diag HK only: This measure is not included in the nominal hk packet and can be included in a diagnostic list
  - All HK modes: This measure is available in both nominal and diagnostic hk.
- Useful size (in bytes):
  - In nominal HK, all measure are transmitted in a 32bits slot.
  - In diag HK, they are transmitted only using the 'useful size' number of bytes.
  - For some measures, we also mention the number of bits that are really useful. DPU can then 'compress' the hk using these numbers.

### 5.3.2 List of Nominal housekeeping Measure

| **0** | **BOLC_HK_1** | | **BOLC** |
|---|---|---|---|
| | SCOS 2000 Display: | DISPLAY_HEX | |
| | Validity at startup: | invalid | |
| | Validity during execution: | Valid only when BOLC is ON and the connection between DMC and BOLC is established | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>BOLC HK entry 1 | | | |

…

| **195** | **BOLC_HK_196** | | **BOLC** |
|---|---|---|---|
| | SCOS 2000 Display: | DISPLAY_HEX | |
| | Validity at startup: | invalid | |
| | Validity during execution: | Valid only when BOLC is ON and the connection between DMC and BOLC is established | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>BOLC HK entry 196 | | | |
| **196** | **DMC_SW_GLOBAL_ST** | | **OBSW** |
| | SCOS 2000 Display: | Bit Field (see description) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (19bits) | |
| Description:<br>DMC Software global status : summary of the status of all the tasks of the software. If an error occurs in any other task, the DMC_SW_ERR will be =1 and the DMC_SW_ERROR will contain a copy of the task error code. Note that these fields remain only for 1 hk packet. | | | |
| bits 0-15 | **DMC_SW_ERROR** | Error code | |
| bit 16 | **DMC_SW_ALIVE** | 1 = DMC OBS is alive<br>0 = DMC OBS is dead (no HK should be received then) | |
| bit 17 | **DMC_SW_ERR** | 1 = Any error in DMC OBS (see bits 0-15 for the error code)<br>0 = No error in DMC OBS | |
| bit 18 | **DMC_SW_COPY_OBS** | 1 = The OBS is being copied in EEPROM right now<br>0 = no copy is being performed now | |
| bits 19-31 | **DMC_SW_SPARE13** | Spare | |
| **197** | **DMC_SEQ_STATUS** | | **SEQ** |
| | SCOS 2000 Display: | Bit Field (see description) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (22bits) | |
| Description:<br>DMC Sequencer status. The Sequencer is the task in charge of the execution of the sequences and trigger commands. This is one of the vital tasks of the DMC OBS. | | | |
| bits 0-15 | **DMC_SEQ_ERROR** | Error code | |
| bit 16 | **DMC_SEQ_TASK_AL** | 1 = Sequencer task is running<br>0 = Sequencer task is not running | |
| bit 17 | **DMC_SEQ_TASK_WR** | 1 = Any error occurred in the Sequencer task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in Sequencer task | |
| bit 18 | **DMC_SEQ_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet | |

| | | |
|---|---|---|
| | | 0 = No error waiting to be signaled |
| bit 19 | **DMC_SEQ_SPARE1** | Spare |
| bit 20 | **DMC_SEQ_IDLE** | 1 = Sequencer in idle mode (no sequence running)<br>0 = Sequencer not in idle mode |
| bit 21 | **DMC_SEQ_RUNNING** | 1 = A sequence is being executed<br>0 = No sequence is being executed |
| bits 22-31 | **DMC_SEQ_SPARE10** | Spare |
| **198** | **DMC_DPU_REC_STAT** | **COM** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (20bits) |

Description:
DPU Receiver status. This taks is in charge of the receptions of commands from DPU and the execution of memory commands (write - dump - check - load). This is one of the vital tasks of the DMC OBS.

| | | |
|---|---|---|
| bits 0-15 | **DMC_DPUR_ERROR** | Error code |
| bit 16 | **DMC_DPUR_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DPUR_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_DPUR_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_DPUR_LINK** | 1 = Waiting connection with DPU<br>0 = Connection established with DPU |
| bits 20-31 | **DMC_DPUR_SPARE12** | Spare |
| **199** | **DMC_DPU_SEN_STAT** | **COM** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (20bits) |

Description:
DPU Sender status. This taks is in charge of the emission of packet to DPU (HK packets, commands ack). This is one of the vital tasks of the DMC OBS.

| | | |
|---|---|---|
| bits 0-15 | **DMC_DPUS_ERROR** | Error code |
| bit 16 | **DMC_DPUS_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DPUS_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_DPUS_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_DPUS_LINK** | 1 = Waiting connection with DPU<br>0 = Connection established with DPU |
| bits 20-31 | **DMC_DPUS_SPARE12** | Spare |
| **200** | **DMC_DECB_REC_STA** | **DEC** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |

| | | | |
|---|---|---|---|
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (22bits) | |

Description:
Blue DEC Receiver task status. This task is receiving the packets from the Blue DEC. In case the Detector simulator has been started; it can also replace the readouts by simulated ones or can even generate readouts if not connected to DEC)

| | | |
|---|---|---|
| bits 0-15 | **DMC_DBR_ERROR** | Error code |
| bit 16 | **DMC_DBR_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DBR_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_DBR_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_DBR_LINK** | 1 = Waiting connection with Blue DEC<br>0 = Connection established with Blue DEC |
| bit 20 | **DMC_DBR_SENDING** | 1 = using simulated readouts<br>0 = using real readouts |
| bit 21 | **DMC_DBR_SIM_TIME** | 1 = using simulated timing (the detector simulator is running and generating the timing)<br>0 = using real timing (valid only when bit 20 is set; replaces the science data received from DEC by simulated readouts) |
| bits 22-31 | **DMC_DBR_SPARE10** | Spare |

| **201** | **DMC_DECB_CTRL_ST** | | **DEC** |
|---|---|---|---|
| | SCOS 2000 Display: | Bit Field (see description) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (21bits) | |

Description:
Blue DEC Controller task status. This task is sending the commands to Blue DEC and control the power on/off.

| | | |
|---|---|---|
| bits 0-15 | **DMC_DBC_ERROR** | Error code |
| bit 16 | **DMC_DBC_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DBC_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_DBC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_DBC_LINK** | 1 = Waiting connection with Blue DEC<br>0 = Connection established with Blue DEC |
| bit 20 | **DMC_DBC_POWER** | 1 = Blue DEC is powered ON<br>0 = Blue DEC is OFF |
| bits 21-31 | **DMC_DBC_SPARE11** | Spare |

| **202** | **DMC_BLUE_PAC_ENC** | | **COM** |
|---|---|---|---|
| | SCOS 2000 Display: | Bit Field (see description) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (20bits) | |

Description:

| | | |
|---|---|---|
| Blue Packet Encoder task status. This task is sending the science packet to the Blue SPU. | | |
| bits 0-15 | **DMC_BPE_ERROR** | Error code |
| bit 16 | **DMC_BPE_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_BPE_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_BPE_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_BPE_LINK** | 1 = Waiting connection with Blue SPU<br>0 = Connection established with Blue SPU |
| bits 20-31 | **DMC_BPE_SPARE12** | Spare |
| **203** | **DMC_DECR_REC_STA** | **DEC** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4(22bits) |
| Description:<br>Red DEC Receiver task status. This task is receiving the packets from the Red DEC. In case the Detector simulator has been started; it can also replace the readouts by simulated ones or can even generate readouts if not connected to DEC) | | |
| bits 0-15 | **DMC_DRR_ERROR** | Error code |
| bit 16 | **DMC_DRR_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DRR_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_DRR_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_DRR_LINK** | 1 = Waiting connection with Red DEC<br>0 = Connection established with Red DEC |
| bit 20 | **DMC_DRR_SENDING** | 1 = using simulated readouts<br>0 = using real readouts |
| bit 21 | **DMC_DRR_SIM_TIME** | 1 = using simulated timing (the detector simulator is running and generating the timing)<br>0 = using real timing (valid only when bit 20 is set; replaces the science data received from DEC by simulated readouts) |
| bits 22-31 | **DMC_DRR_SPARE10** | Spare |
| **204** | **DMC_DECR_CTRL_ST** | **DEC** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (21bits) |
| Description:<br>Red DEC Controller task status. This task is sending the commands to Red DEC and control the power on/off. | | |
| bits 0-15 | **DMC_DRC_ERROR** | Error code |
| bit 16 | **DMC_DRC_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_DRC_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless |

| | | | |
|---|---|---|---|
| | | bit 18 is set)<br>0 = No error in this task | |
| bit 18 | **DMC_DRC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled | |
| bit 19 | **DMC_DRC_LINK** | 1 = Waiting connection with Red DEC<br>0 = Connection established with Red DEC | |
| bit 20 | **DMC_DRC_POWER** | 1 = Red DEC is powered ON<br>0 = Red DEC is OFF | |
| bits 21-31 | **DMC_DRC_SPARE11** | Spare | |
| **205** | **DMC_RED_PAC_ENC** | | **COM** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit Field (see description)<br>Valid<br>Always Valid<br>All HK modes<br>4 (20bits) | |
| Description:<br>Red Packet Encoder task status. This task is sending the science packet to the Red SPU. | | | |
| bits 0-15 | **DMC_RPE_ERROR** | Error code | |
| bit 16 | **DMC_RPE_TASK_AL** | 1 = this task is running<br>0 = this task is not running | |
| bit 17 | **DMC_RPE_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task | |
| bit 18 | **DMC_RPE_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled | |
| bit 19 | **DMC_RPE_LINK** | 1 = Waiting connection with Red SPU<br>0 = Connection established with Red SPU | |
| bits 20-31 | **DMC_RPE_SPARE12** | Spare | |
| **206** | **DMC_BOL_REC_STAT** | | **BOLC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit Field (see description)<br>Valid<br>Always Valid<br>All HK modes<br>4 (22bits) | |
| Description:<br>BOL Receiver task status. This task is receiving the packets from the BOLC. In case the Detector simulator has been started; it can also replace the readouts by simulated ones or can even generate readouts if not connected to BOLC) | | | |
| bits 0-15 | **DMC_BR_ERROR** | Error code | |
| bit 16 | **DMC_BR_TASK_AL** | 1 = this task is running<br>0 = this task is not running | |
| bit 17 | **DMC_BR_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task | |
| bit 18 | **DMC_BR_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled | |
| bit 19 | **DMC_BR_LINK** | 1 = Waiting connection with Red DEC<br>0 = Connection established with Red DEC | |
| bit 20 | **DMC_BR_SENDING** | 1 = using simulated readouts<br>0 = using real readouts | |

| | | |
|---|---|---|
| bit 21 | **DMC_BR_SIM_TIME** | 1 = using simulated timing (the detector simulator is running and generating the timing)<br>0 = using real timing (valid only when bit 20 is set; replaces the science data received from DEC by simulated readouts) |
| bits 22-31 | **DMC_BR_SPARE10** | Spare |
| **207** | **DMC_BOL_CTRL_STA** | **BOLC** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (20bits) |
| Description:<br>BOL Controller task status. This task is sending the commands to BOLC. | | |
| bits 0-15 | **DMC_BC_ERROR** | Error code |
| bit 16 | **DMC_BC_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_BC_TASK_WR** | 1 = Any error occurred in this task, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_BC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_BC_LINK** | 1 = Waiting connection with Red DEC<br>0 = Connection established with Red DEC |
| bits 20-31 | **DMC_BC_SPARE12** | Spare |
| **208** | **DMC_GRAT_CTRL_ST** | **GRAT** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Bits 16, 19 and 21-31 are updated only while the Grating is powered on |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Grating Controller status. | | |
| bits 0-15 | **DMC_GC_ERROR** | Error code |
| bit 16 | **DMC_GC_LL_SC** | Spare (will be Launch Lock Short-Circuit) |
| bit 17 | **DMC_GC_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_GC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_GC_LL_MOVING** | 1 = Launch lock is moving<br>0 = Launch lock is not moving |
| bit 20 | **DMC_GC_POWER** | 1 = Grating is powered ON<br>0 = Grating is powered OFF |
| bit 21 | **DMC_GC_PID** | 1 = Grating controller is enabled<br>0 = Grating controller is disabled |
| bit 22 | **DMC_GC_COMMUT** | 1 = Grating controller commutation is enabled (obsolete)<br>0 = Grating controller commutation is disabled |
| bit 23 | **DMC_GC_LS** | 1 = Limit switch is pressed<br>0 = Limit switch is not pressed<br>Remark: During a homing, the grating will stay only around 1 second on the limit switch. So, it might happen that you do not |

| | | |
|---|---|---|
| | | see this bit set to 1 in the nominal HK. |
| bit 24 | **DMC_GC_DEGRADED** | 1 = Grating is in degraded mode<br>0 = Grating is in nominal mode |
| bit 25 | **DMC_GC_UP** | 1 = Moving UP<br>0 = Not moving UP |
| bit 26 | **DMC_GC_DOWN** | 1 = Moving DOWN<br>0 = Not moving DOWN |
| bit 27 | **DMC_GC_SYNCHRO** | 1 = Using synchro (movement starts only right after a synchro pulse)<br>0 = Not using synchro (movements starts anytime) |
| bit 28 | **DMC_GC_HOM_PROG** | 1 = Homing is in progress<br>0 = No homing in progress (not started or completed) |
| bit 29 | **DMC_GC_HOM_COMP** | 1 = Homing has completed<br>0 = Homing has not (yet) completed |
| bit 30 | **DMC_GC_LL_LOCKED** | 1 = Launch lock is locked<br>0 = Launch lock is not locked |
| bit 31 | **DMC_GC_LL_UNLOCKED** | 1 = Launch lock is unlocked<br>0 = Launch lock is not unlocked<br>Remark : The Lauch lock position indicators are powered only when the launch lock actuator(s) is(are) powered. Status is therefore only valid during that time (nominally 5 sec., i.e. for 2 or 3 nominal HK packets maximum) |
| **209** | **DMC_CHOP_CTRL_ST** | **CHOP** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (28bits) |
| Description:<br>Chopper Controller status. | | |
| bits 0-15 | **DMC_CC_ERROR** | Error code |
| bit 16 | **DMC_CC_SPARE1A** | Spare |
| bit 17 | **DMC_CC_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_CC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_CC_SPARE1B** | Spare |
| bit 20 | **DMC_CC_POWER** | 1 = Chopper is powered ON<br>0 = Chopper is OFF |
| bit 21 | **DMC_CC_PID** | 1 = Chopper controller is enabled<br>0 = Chopper controller is disabled |
| bit 22 | **DMC_CC_COMMUT** | 1 = Chopper controller commutation is enabled<br>0 = Chopper controller commutation is disabled |
| bit 23 | **DMC_CC_LOOP** | 1 = Open loop mode<br>0 = Closed loop mode |
| bit 24 | **DMC_CC_SPARE1C** | spare |
| bit 25 | **DMC_CC_UP** | 1 = Moving UP<br>0 = Not moving UP |
| bit 26 | **DMC_CC_DOWN** | 1 = Moving DOWN<br>0 = Not moving DOWN |
| bit 27 | **DMC_CC_SYNCHRO** | 1 = Using synchro (movement starts only right after a synchro pulse)<br>0 = Not using synchro (movements starts anytime) |
| bits 28-31 | **DMC_CC_SPARE4** | Spare |
| **210** | **DMC_FW_SPEC_CTRL** | **FW** |

| | | |
|---|---|---|
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (30bits) |
| Description:<br>Filter Wheel Spectro Controller status. | | |
| bits 0-15 | **DMC_FWSC_ERROR** | Error code |
| bit 16 | **DMC_FWSC_SPARE1A** | Spare |
| bit 17 | **DMC_FWSC_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_FWSC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_FWSC_SPARE1B** | Spare |
| bit 20 | **DMC_FWSC_POWER** | 1 = Filter Wheel Spectro is powered ON<br>0 = Filter Wheel Spectro is OFF |
| bit 21-24 | **DMC_FWSC_SPARE4** | Spare |
| bit 25 | **DMC_FWSC_MOVING** | 1 = Currently moving (actually = FW controller is currently sending current in the coils)<br>0 = Currenlty not moving |
| bit 26 | **DMC_FWSC_SEARCH_A** | 1 = Searching position A<br>0 = Not searching position A |
| bit 27 | **DMC_FWSC_SEARCH_B** | 1 = Searching position B<br>0 = Not searching position B |
| bit 28 | **DMC_FWSC_POS_A** | 1 = Currently at position A<br>0 = Currenlty not at position A |
| bit 29 | **DMC_FWSC_POS_B** | 1 = Currently at position B<br>0 = Currenlty not at position B |
| bit 30 | **DMC_FWSC_POSC_A** | 1 = Control threshold has been reached for position A<br>0 = sensor < control threshold for position A |
| bit 31 | **DMC_FWSC_POSC_B** | 1 = Control threshold has been reached for position B<br>0 = sensor < control threshold for position B |
| **211** | **DMC_FW_PHOT_CTRL** | **FW** |
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (30bits) |
| Description:<br>Filter Wheel Photo Controller status. | | |
| bits 0-15 | **DMC_FWPC_ERROR** | Error code |
| bit 16 | **DMC_FWPC_SPARE1A** | Spare |
| bit 17 | **DMC_FWPC_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_FWPC_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_FWPC_SPARE1B** | Spare |
| bit 20 | **DMC_FWPC_POWER** | 1 = Filter Wheel Photo is powered ON<br>0 = Filter Wheel Photo is OFF |
| bit 21-24 | **DMC_FWPC_SPARE4** | Spare |
| bit 25 | **DMC_FWPC_MOVING** | 1 = Currently moving (actually = FW controller is currently sending current in the coils) |

| | | |
|---|---|---|
| | | 0 = Currenlty not moving |
| bit 26 | **DMC_FWPC_SEARCH_A** | 1 = Searching position A<br>0 = Not searching position A |
| bit 27 | **DMC_FWPC_SEARCH_B** | 1 = Searching position B<br>0 = Not searching position B |
| bit 28 | **DMC_FWPC_POS_A** | 1 = Currently at position A<br>0 = Currenlty not at position A |
| bit 29 | **DMC_FWPC_POS_B** | 1 = Currently at position B<br>0 = Currenlty not at position B |
| bit 30 | **DMC_FWSC_POSC_A** | 1 = Control threshold has been reached for position A<br>0 = sensor < control threshold for position A |
| bit 31 | **DMC_FWSC_POSC_B** | 1 = Control threshold has been reached for position B<br>0 = sensor < control threshold for position B |

| **212** | **DMC_CHECKSUM** | **SPARE** |
|---|---|---|
| | SCOS 2000 Display: | Hex |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |

Description:
The checksum on the hk packet. Computed this way:

```
crc=0xFFFFFFFF;
for (i = 2; i < DMC_SPARE_3; i++) {crc = Crc32(gHkPacketBuffer[i], crc);}
for (i = DMC_SPARE_3+1; i < length; i++) {crc=Crc32(gHkPacketBuffer[i], crc);}
```

| **213** | **DMC_CS1_CTRL_STA** | **CS** |
|---|---|---|
| | SCOS 2000 Display: | Bit Field (see description) |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 (28bits) |

Description:
Calibratin Source 1 Controller status (currently not implemented).

| | | |
|---|---|---|
| bits 0-15 | **DMC_CS1C_ERROR** | Error code |
| bit 16 | **DMC_CS1C_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_CS1C_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_CS1C_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_CS1C_SPARE1** | Spare |
| bit 20 | **DMC_CS1C_POWER** | 1 = Calibration Source 1 is powered ON<br>0 = Calibration Source 1 is OFF |
| bit 21 | **DMC_CS1C_PID** | 1 = Calibration Source 1 controller is enabled<br>0 = Calibration Source 1 controller is disabled |
| bit 22 | **DMC_CS1C_COMMUT** | 1 = Calibration Source 1 controller commutation is enabled<br>0 = Calibration Source 1 controller commutation is disabled |
| bit 23 | **DMC_CS1C_LOOP** | 1 = Cpen loop mode<br>0 = Closed loop mode |
| bit 24 | **DMC_CS1C_SPARE1B** | spare |
| bit 25 | **DMC_CS1C_UP** | 1 = Moving UP<br>0 = Not moving UP |
| bit 26 | **DMC_CS1C_DOWN** | 1 = Moving DOWN<br>0 = Not moving DOWN |
| bit 27 | **DMC_CS1C_SYNCHRO** | 1 = Using synchro (movement starts only right after a synchro pulse)<br>0 = Not using synchro (movements starts anytime) |

| | | | |
|---|---|---|---|
| bits 28-31 | **DMC_CS1C_SPARE4** | Spare | |
| **214** | **DMC_CS2_CTRL_STA** | | **CS** |
| | SCOS 2000 Display: | Bit Field (see description) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (28bits) | |
| Description:<br>Calibratin Source 2 Controller status (currently not implemented). | | | |
| bits 0-15 | **DMC_CS2C_ERROR** | Error code | |
| bit 16 | **DMC_CS2C_TASK_AL** | 1 = this task is running<br>0 = this task is not running | |
| bit 17 | **DMC_CS2C_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task | |
| bit 18 | **DMC_CS2C_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled | |
| bit 19 | **DMC_CS2C_SPARE1** | Spare | |
| bit 20 | **DMC_CS2C_POWER** | 1 = Calibration Source 2 is powered ON<br>0 = Calibration Source 2 is OFF | |
| bit 21 | **DMC_CS2C_PID** | 1 = Calibration Source 2 controller is enabled<br>0 = Calibration Source 2 controller is disabled | |
| bit 22 | **DMC_CS2C_COMMUT** | 1 = Calibration Source 2 controller commutation is enabled<br>0 = Calibration Source 2 controller commutation is disabled | |
| bit 23 | **DMC_CS2C_LOOP** | 1 = Cpen loop mode<br>0 = Closed loop mode | |
| bit 24 | **DMC_CS2C_SPARE1B** | spare | |
| bit 25 | **DMC_CS2C_UP** | 1 = Moving UP<br>0 = Not moving UP | |
| bit 26 | **DMC_CS2C_DOWN** | 1 = Moving DOWN<br>0 = Not moving DOWN | |
| bit 27 | **DMC_CS2C_SYNCHRO** | 1 = Using synchro (movement starts only right after a synchro pulse)<br>0 = Not using synchro (movements starts anytime) | |
| bits 28-31 | **DMC_CS2C_SPARE4** | Spare | |
| **215** | **DMC_SEQ_OPTIONS** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 1 (4bits) | |
| Description:<br>Sequencer options : select the synchronization signal used to execute the sequences | | 1 = synchronize on Blue DEC ramps<br>2 = synchronize on Red DEC ramps<br>4 = synchronize on BOL readouts | |
| **216** | **DMC_SEQ_POINTER** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (8bits) | |
| Description:<br>Sequence Pointer : Indicates the current position in the sequence (0 based index showing the 'line number' in the sequence) | | | |

| | | | |
|---|---|---|---|
| **217** | **DMC_SEQ_LOOP_ID0** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sequence Loop 0 Index : Current index in the highest level loop (decrementing counter gives the number of iteration left) | | | |
| **218** | **DMC_SEQ_LOOP_ID1** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>… | | | |
| **219** | **DMC_SEQ_LOOP_ID2** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>… | | | |
| **220** | **DMC_SEQ_LOOP_ID3** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>… | | | |
| **221** | **DMC_SEQ_LOOP_ID4** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sequence Loop 4 Index : Current index in the lowest level loop | | | |
| **222** | **DMC_SEQ_WAIT_IND** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current index in the current 'WAIT' command (decrementing counter gives the number of wait remaining) | | | |
| **223** | **DMC_SEQ_LABEL** | | **SEQ** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 1 | |
| Description:<br>Sequence label | | | |
| **224** | **DMC_OBSID** | | **TIME** |
| | SCOS 2000 Display: | ? | |

| | | | |
|---|---|---|---|
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>OBSID | | | |
| **225** | **DMC_BBID** | | **TIME** |
| | SCOS 2000 Display: | ? | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>BBID | | | |
| **226** | **DMC_TIME_1** | | **TIME** |
| | SCOS 2000 Display: | ? | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>Contains the seconds of the time. (number which relects the amount of seconds elapsed since 1-Jan-1958 0:00:00 UT). | | | |
| **227** | **DMC_TIME_2** | | **TIME** |
| | SCOS 2000 Display: | ? | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Contains the fraction of seconds of the time in 65536th units. | | | |
| **228** | **DMC_DECB_REC_PAC** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Blue DEC Receiver packet counter. Counts the number of packets received from Blue DEC since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing). These are the 16 lsb of CRDC in science header | | | |
| **229** | **DMC_DECR_REC_PAC** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Red DEC Receiver packet counter. Counts the number of packets received from Red DEC since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing).These are the 16 lsb of CRDC in science header | | | |
| **230** | **DMC_DECB_CTRL_PA** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Blue DEC Controller packet counter. Counts the number of packets (commands) sent to Blue DEC since the | | | |

software has been started (since it is only 16 bits, it is only meant to see that it is incrementing).

| | | | |
|---|---|---|---|
| **231** | **DMC_DECR_CTRL_PA** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
Red DEC Controller packet counter. Counts the number of packets (commands) sent to Red DEC since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing).

| | | | |
|---|---|---|---|
| **232** | **DMC_BLUE_ENC_PAC** | | **COM** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
Blue Packet Encoder packet counter. Counts the number of packet sent to Blue SPU since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing)

| | | | |
|---|---|---|---|
| **233** | **DMC_RED_ENC_PAC** | | **COM** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
Red Packet Encoder packet counter. Counts the number of packet sent to Red SPU since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing)

| | | | |
|---|---|---|---|
| **234** | **DMC_BOL_REC_PAC** | | **BOLC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
BOL Receiver packet counter. Counts the number of packets received from BOLC since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing)

| | | | |
|---|---|---|---|
| **235** | **DMC_BOL_CTRL_PAC** | | **BOLC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
BOL Controller packet counter. Counts the number of packets (commands) sent to BOLC since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing).

| | | | |
|---|---|---|---|
| **236** | **DMC_DPU_REC_PAC** | | **COM** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
DPU Receiver packet counter. Counts the number of packets received from DPU since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing).

| | | | |
|---|---|---|---|
| **237** | **DMC_DPU_SEND_PAC** | | **COM** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |

| | | |
|---|---|---|
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 |
| Description:<br>DPU Sender packet counter. Counts the number of packets sent to DPU since the software has been started (since it is only 16 bits, it is only meant to see that it is incrementing). | | |
| **238** | **DMC_B_SPEC_READ** | **DEC** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Counts the readout from Blue DEC received by the DEC/MEC since the last setting of Time. | | |
| **239** | **DMC_R_SPEC_READ** | **DEC** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Counts the readout from Red DEC received by the DEC/MEC since the last setting of Time. | | |
| **240** | **DMC_SYNC_COUNT** | **SYNC** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Counts the currently selected synchro. This counter is positioned before the delay circuit while the CRDCCP counter is positioned after the delay circuit. | | |
| **241** | **DMC_CPU_LOAD** | **OBSW** |
| | SCOS 2000 Display: | Decimal : divide value by 10 to get the percents |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | Nominal HK Only |
| | Useful size (in bytes): | 2 (10bits) |
| Description:<br>Cpu workload : Values between [0, 1000]. Each unit represents 0.1% | | |
| **242** | **DMC_IRS_CNT** | **OBSW** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Servo IRQ Counter : Counts the number of call to the interrupt routine containing the Chopper Controller, the Grating Controller, the Filter Wheel Controller and the analog HK acquisition routine. This should increment by 8192/sec in spectro and 8320/sec in photo with nominal configuration of the timing FPGA | | |
| **243** | **DMC_VID** | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal |
| | Validity at startup: | Valid |
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 4 |
| Description:<br>Software Version Number : Note that this variable is only modifiable by patching (and not by a write command). The MSB is the main version number and the 3 other bytes are used for intermediate numbering. (example: 0x05020300 for version 5.2.3.0) | | |
| **244** | **DMC_CHOP_CUR_POS** | **CHOP** |

| | | | |
|---|---|---|---|
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/- 32767=+/- 10V)<br>Valid<br>Always Valid<br>All HK modes<br>4 (16bits) | |
| Description:<br>Chopper : actual position. This is corresponding to the Field Plates output voltage with an amplification gain of 50.243 (!valid only for QM DEC/MEC!) | | | |
| **245** | **DMC_CHOP_SETPOIN** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only while chopper controller is enabled<br>All HK modes<br>4 (16bits) | |
| Description:<br>Chopper : position servo setpoint. Same units as position (DMC_CHOP_CUR_POS). | | | |
| **246** | **DMC_CHOP_TARGET** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only while chopper controller is enabled<br>All HK modes<br>4 (16bits) | |
| Description:<br>Chopper : final position for move. Same units as position (DMC_CHOP_CUR_POS). | | | |
| **247** | **DMC_CHOP_PID_ERR** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only while chopper controller is enabled<br>All HK modes<br>4 (16bits) | |
| Description:<br>Chopper : current error between position and setpoint (=DMC_CHOP_SETPOINT-DMC_CHOP_CUR_POS). Same units as position (DMC_CHOP_CUR_POS) | | | |
| **248** | **DMC_CHOP_PID_ACC** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (to be displayed in raw values)<br>Invalid<br>Valid only while chopper controller is enabled<br>All HK modes<br>4 (32bits) | |
| Description:<br>Chopper : integral accumulator of servo PID algorithm. Same units as position (DMC_CHOP_CUR_POS) | | | |
| **249** | **DMC_CHOP_MAX_DIT** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Valid<br>Always Valid<br>All HK modes<br>4 (16bits) | |
| Description:<br>Chopper : Maximum Dither Value. Same units as position (DMC_CHOP_CUR_POS) | | | |
| **250** | **DMC_GRAT_CUR_POS** | | **GRAT** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal or deg min sec<br>Invalid<br>Valid only while Grating is powered on<br>All HK modes<br>4 (24bits) | |
| Description:<br>Grating : actual position from readout by HK, in arbitrary units ( signed int ). 1 unit = (360*3600)/(128*65536) = 0.1545 arcsec. | | | |
| **251** | **DMC_GRAT_SETPOIN** | | **GRAT** |

| | | | |
|---|---|---|---|
| | SCOS 2000 Display: | Decimal or deg min sec | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating controller is enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (24bits) | |
| Description:<br>Grating : position servo setpoint, same units as actual position (DMC_GRAT_CUR_POS) | | | |
| **252** | **DMC_GRAT_TARGET** | | **GRAT** |
| | SCOS 2000 Display: | Decimal or deg min sec | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating controller is enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (24bits) | |
| Description:<br>Grating : final position for move, same units as actual position (DMC_GRAT_CUR_POS) | | | |
| **253** | **DMC_GRAT_PID_ERR** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating controller is enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 (24bits) | |
| Description:<br>Grating : current error between position and setpoint (=DMC_GRAT_SETPOINT-DMC_GRAT_CUR_POS). Same units as actual position (DMC_GRAT_CUR_POS). | | | |
| **254** | **DMC_GRAT_PID_ACC** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating controller is enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>Grating : integral accumulator of servo PID algorithm. Same units as actual position (DMC_GRAT_CUR_POS). | | | |
| **255** | **DMC_FWSP_CUR_POS** | | **FW** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | It is updated only while the FW is powered on. | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 1 (4bits) | |
| Description:<br>FW Spectro current position:<br>-1 : unknown (between the two position or the FW has not been powered on yet so we don't know where it is)<br>0 : position A<br>1 : position B | | | |
| **256** | **DMC_FWGRAT_HALLA** | | **FW** |
| | SCOS 2000 Display: | Decimal (for grating hall sensors: +/-32767 = +/-49mV, for FW hall sensors: +/-32767 = +/-102mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while FW or Grating is powered on | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Value of the Hall A sensor of the currently active mechanism (FW or grating). Output voltage of the hall sensor under 0.7 mA excitation with an amplifier gain of 206. (+32767 = +10 V and -32767 = -10 V) | | | |
| **257** | **DMC_FWGRAT_HALLB** | | **FW** |
| | SCOS 2000 Display: | Decimal (for grating hall sensors: +/-32767 = +/-49mV, for FW hall sensors: +/-32767 = +/-102mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while FW or Grating is powered on | |
| | Available in: | All HK modes | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 2 | |
| Description:<br>Value of the Hall B sensor of the currently active mechanism (FW or grating). Output voltage of the hall sensor under 0.7 mA excitation with an amplifier gain of 206. (+32767 = +10 V and -32767 = -10 V) | | | |
| **258** | **DMC_CHOP_OUTPUT** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767 = +/-130 mA)<br>Invalid<br>Valid only while chopper controller is enabled<br>All HK modes<br>4 | |
| Description:<br>Current commanded in chopper coils = output of the controller. | | | |
| **259** | **DMC_ISR_STAT** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit field<br>Valid<br>Always Valid<br>All HK modes<br>4 (2bits) | |
| Description:<br>Interrupt routine status | | | |
| bit 0 | **DMC_ISR_SPARE_1** | Don't care | |
| bit 1 | **DMC_ISR_SYNC_RES** | 1 = sync received (reset to zero when software 'consumes' it) TBC<br>0 = no sync | |
| bits 2 - 31 | **DMC_ISR_SPARE_30** | Spares | |
| **260** | **DMC_FWPH_CUR_POS** | | **FW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>It is updated only while the FW is powered on.<br>All HK modes<br>1 (4bits) | |
| Description:<br>FW Photo current position:<br>-1 : unknown (between the two position or the FW has not been powered on yet so we don't know where it is)<br>0 : position A<br>1 : position B | | | |
| **261** | **DMC_SPARE1** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Always Invalid<br>All HK modes<br>2 | |
| Description:<br>spare | | | |
| **262** | **DMC_SPARE2** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Always Invalid<br>All HK modes<br>2 | |
| Description:<br>spare | | | |
| **263** | **DMC_PLL_RES_LO** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in: | Decimal (2^32 = 1 period) should be interpreted as a signed number<br>Valid<br>Valid only when locked on OBT or BOLC (not valid in free run)<br>All HK modes | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 4 | |
| Description:<br>Timing FPGA PLL residue (Low word). Phase difference measured at each period of the selected synchronization signal. | | | |
| Limit checking :<br>See section 'adjusting the timing parameters' | | | |
| **264** | **DMC_PLL_RES_HI** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Valid<br>Valid only when locked on OBT or BOLC (not valid in free run)<br>All HK modes<br>2 | |
| Description:<br>Timing FPGA PLL residue (High word). Number of reference frequency periods | | | |
| **265** | **DMC_DECB_VDDD_3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VDDD Voltage Power Supply Group 3 | | | |
| Limit checking :<br>See section 'adjusting the timing parameters' | | | |
| **266** | **DMC_DECB_VSS_3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VSS Voltage Power Supply Group 3 | | | |
| **267** | **DMC_DECB_VGND_3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>GND Voltage Power Supply Group 3 | | | |
| **268** | **DMC_DECB_VCAN1_3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N1 Voltage Power Supply Group 3 | | | |
| **269** | **DMC_DECB_VCAN2_3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N2 Voltage Power Supply Group 3 | | | |
| **270** | **DMC_DECB_V0BIAS3** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup: | Decimal (0 = -6V, 65535 = 6V)<br>Invalid | |

| | | | |
|---|---|---|---|
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Zero Bias Voltage Power Supply Group 3 | | | |
| **271** | **DMC_DECB_VBI_R_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Bias R Voltage Power Supply Group 3 | | | |
| **272** | **DMC_DECB_V0V_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>0V reference Voltage Power Supply Group 3 | | | |
| **273** | **DMC_DECB_VSCP_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Cascode P Voltage Power Supply Group 3 | | | |
| **274** | **DMC_DECB_VDDR_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current Mirror Voltage Power Supply Group 3 | | | |
| **275** | **DMC_DECB_VDDA_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDA Voltage Power Supply Group 3 | | | |
| **276** | **DMC_DECB_VWELL_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Well Voltage Power Supply Group 3 | | | |
| **277** | **DMC_DECB_IDDA_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

| | | | |
|---|---|---|---|
| Description: VDDA Current Power Supply Group 3 | | | |
| **278** | **DMC_DECB_IDDD_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: VDDD Current Power Supply Group 3 | | | |
| **279** | **DMC_DECB_ISS_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: VSS Current Power Supply Group 3 | | | |
| **280** | **DMC_DECB_IGND_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: GND Current Power Supply Group 3 | | | |
| **281** | **DMC_DECB_HEAT_C** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -25mA, 32767 = 25mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Blue DEC Heater Current | | | |
| **282** | **DMC_DECB_HEAT_V** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -37.5V, 32767 = 37.5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Heater Voltage | | | |
| **283** | **DMC_DECB_REF_0V3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -5V, 32767 = 5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: 0V Reference voltage for DEC Base Group 3 ADC | | | |
| **284** | **DMC_DECB_DCDC_T3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 100Kohms) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: DC/DC temperature The temperature can be computed with: $T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3)$ | | | |

Where a0 = 1,2835e-3
a1 = 2,3646e-4
a3 = 9,1416e-8
Limit checking :
Warning when T(K) out of [243.15, 343.15]
Switch-off DMC when T(K) out of [218.15, 353.15]

| | | | |
|---|---|---|---|
| **285** | **DMC_DECB_SPARE5** | | **DEC** |
| | SCOS 2000 Display: | none | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
spare

| | | | |
|---|---|---|---|
| **286** | **DMC_DECB_DCDC_P5** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-1700mA, 32767=1700mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
Current from +5V power supply

| | | | |
|---|---|---|---|
| **287** | **DMC_DECB_AC_CUR** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-349.57mA, 32767=349.57mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
AC Current

| | | | |
|---|---|---|---|
| **288** | **DMC_DECB_TS_ST_3** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (4bits) | |

Description:
Bit field showing the status of each of the detector array temperature sensors. 2bits for each sensors:

| | | |
|---|---|---|
| 00 | Sensor inactive (measure is invalid) | |
| 01 | Measure has been done using Low Gain Current Measure | |
| 10 | Measure has been done using High Gain Current Measure | |
| 11 | Error in measure (measure is invalid) | |
| bits 0-1 | **DMC_DECB_TS_1_ST_3** | temperature sensor 1 status |
| bits 2-3 | **DMC_DECB_TS_2_ST_3** | temperature sensor 2 status |
| bits 4-15 | **DMC_DECB_TS_SP_3** | spare |

| | | | |
|---|---|---|---|
| **289** | **DMC_DECB_CL_RO_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
Number of CRE clocks per readout

| | | | |
|---|---|---|---|
| **290** | **DMC_DECB_RO_RA_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |

| | | | |
|---|---|---|---|
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of readouts per ramp | | | |
| **291** | **DMC_DECB_CR_ST_3** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>CRE group 3 status word. Note, bit 0-14 are command readback. Bit 15 is the real status of the CRE power. | | | |
| bit 0 | **DMC_DECB_CR3_ST_POW** | 1 = CRE power on command readback<br>0 = CRE power off command readback | |
| bit 1 | **DMC_DECB_CR3_ST_SEL** | 1 = CRE Active<br>0 = CRE inactive | |
| bit 2-3 | **DMC_DECB_CR3_ST_CS** | Capacitor select read back<br>00 = 100fF<br>10 = 200fF<br>01 = 400fF<br>11 = 1pF | |
| bit 4 | **DMC_DECB_CR3_ST_CUR** | 1 = curing<br>0 = not curing | |
| bit 5 | **DMC_DECB_CR3_ST_SP1** | Spare | |
| bit 6 | **DMC_DECB_CR3_ST_SIM** | 1 = simulation mode<br>0 = nominal mode | |
| bit 7 | **DMC_DECB_CR3_ST_TE** | 1 = temperature sensors enabled<br>0 = temperature sensors disabled | |
| bit 8 | **DMC_DECB_CR3_ST_NDS** | 1 = Non destructive sync is 2 CRE Clock width<br>0 = Non destructive sync is 1 CRE Clock width | |
| bit 9 | **DMC_DECB_CR3_ST_RA** | 1 = ramp simulation ON<br>0 = ramp simulation OFF | |
| bit 10 | **DMC_DECB_CR3_ST_FL** | 1 = Flasher is ON<br>0 = Flasher is OFF | |
| bit 11 | **DMC_DECB_CR3_ST_HE** | 1 = Heater is ON<br>0 = Heater is OFF | |
| bit 12-14 | **DMC_DECB_CR3_ST_SP2** | Spare | |
| bit 15 | **DMC_DECB_CR3_ST_CRPO<br>W** | 1 = CRE powered on<br><br>0 = CRE powered off | |
| **292** | **DMC_DECB_BR_CM_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal ( 0 = 0V, 4095 = +1V ) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (12bits) | |
| Description:<br>Bias R command readback | | | |
| **293** | **DMC_DECB_ZB_CM_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal ( 0 = 0V, 4095 = +1V ) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (12bits) | |
| Description:<br>Zero Bias command readback | | | |
| **294** | **DMC_DECB_SR_RB_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |

| | | | |
|---|---|---|---|
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>simulation register readback | | | |
| **295** | **DMC_DECB_TS_1_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (1 unit = 1 ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Temperature Sensor 1 resistor value | | | |
| **296** | **DMC_DECB_TS_2_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal (1 unit = 1 ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Temperature Sensor 2 resistor value | | | |
| **297** | **DMC_DECB_RO_CO_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>readout ID, counts from readouts_per_ramp-1 to 0 | | | |
| **298** | **DMC_DECB_RA_CO_3** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>ramp counter, increments until reset | | | |
| **299** | **DMC_DECB_VDDD_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDD Voltage Power Supply Group 4 | | | |
| **300** | **DMC_DECB_VSS_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VSS Voltage Power Supply Group 4 | | | |
| **301** | **DMC_DECB_VGND_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 2 | |
| Description:<br>GND Voltage Power Supply Group 4 | | | |
| **302** | **DMC_DECB_VCAN1_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N1 Voltage Power Supply Group 4 | | | |
| **303** | **DMC_DECB_VCAN2_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N2 Voltage Power Supply Group 4 | | | |
| **304** | **DMC_DECB_V0BIAS4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Zero Bias Voltage Power Supply Group 4 | | | |
| **305** | **DMC_DECB_VBI_R_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Bias R Voltage Power Supply Group 4 | | | |
| **306** | **DMC_DECB_V0V_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>OV reference Voltage Power Supply Group 4 | | | |
| **307** | **DMC_DECB_VSCP_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode P Voltage Power Supply Group 4 | | | |
| **308** | **DMC_DECB_VDDR_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Current Mirror Voltage Power Supply Group 4 | | | |

| | | | |
|---|---|---|---|
| **309** | **DMC_DECB_VDDA_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDA Voltage Power Supply Group 4 | | | |
| **310** | **DMC_DECB_VWELL_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Well Voltage Power Supply Group 4 | | | |
| **311** | **DMC_DECB_IDDA_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDA Current Power Supply Group 4 | | | |
| **312** | **DMC_DECB_IDDD_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDD Current Power Supply Group 4 | | | |
| **313** | **DMC_DECB_ISS_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VSS Current Power Supply Group 4 | | | |
| **314** | **DMC_DECB_IGND_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>GND Current Power Supply Group 4 | | | |
| **315** | **DMC_DECB_FLASH_C** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -25mA, 32767 = 25mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Dec Blue Flasher Current | | | |
| **316** | **DMC_DECB_FLASH_V** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -37.5V, 32767 = 37.5V) | |
| | Validity at startup: | Invalid | |

| | | | |
|---|---|---|---|
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Dec Blue Flasher Voltage | | | |
| **317** | **DMC_DECB_REF_0V4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -5V, 32767 = 5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: 0V Reference voltage for DEC Base Group 4 ADC | | | |
| **318** | **DMC_DECB_DCDC_T4** | | **DEC** |
| | SCOS 2000 Display: | Decimal (TBD) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: DC/DC temperature (not connected, spare sensor) | | | |
| **319** | **DMC_DECB_SPARE5B** | | **DEC** |
| | SCOS 2000 Display: | none | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **320** | **DMC_DECB_DCDC_P15** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-144mA, 32767=144mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Current from +15V power supply | | | |
| **321** | **DMC_DECB_DCDC_N15** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-144mA, 32767=144mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: | | | |
| Current from -15V power supply | | | |
| **322** | **DMC_DECB_TS_ST_4** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (4bits) | |
| Description: Bit field showing the status of each of the detector array temperature sensors. 2bits for each sensors: | | | |
| 00 | Sensor inactive (measure is invalid) | | |
| 01 | Measure has been done using Low Gain Current Measure | | |
| 10 | Measure has been done using High Gain Current Measure | | |

| | | | |
|---|---|---|---|
| 11 | Error in measure (measure is invalid) | | |
| bits 0-1 | **DMC_DECB_TS_1_ST_4** | temperature sensor 1 status | |
| bits 2-3 | **DMC_DECB_TS_2_ST_4** | temperature sensor 2 status | |
| bits 4-15 | **DMC_DECB_TS_SP_4** | spare | |
| **323** | **DMC_DECB_CL_RO_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of CRE clocks per readout | | | |
| **324** | **DMC_DECB_RO_RA_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of readouts per ramp | | | |
| **325** | **DMC_DECB_CR_ST_4** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>CRE group 4 status word. Note, bit 0-14 are command readback. Bit 15 is the real status of the CRE power. | | | |
| bit 0 | **DMC_DECB_CR4_ST_POW** | 1 = CRE power on command readback<br>0 = CRE power off command readback | |
| bit 1 | **DMC_DECB_CR4_ST_SEL** | 1 = CRE Active<br>0 = CRE inactive | |
| bit 2-3 | **DMC_DECB_CR4_ST_CS** | Capacitor select read back<br>00 = 100fF<br>10 = 200fF<br>01 = 400fF<br>11 = 1pF | |
| bit 4 | **DMC_DECB_CR4_ST_CUR** | 1 = curing<br>0 = not curing | |
| bit 5 | **DMC_DECB_CR4_ST_SP1** | Spare | |
| bit 6 | **DMC_DECB_CR4_ST_SIM** | 1 = simulation mode<br>0 = nominal mode | |
| bit 7 | **DMC_DECB_CR4_ST_TE** | 1 = temperature sensors enabled<br>0 = temperature sensors disabled | |
| bit 8 | **DMC_DECB_CR4_ST_NDS** | 1 = Non destructive sync is 2 CRE Clock width<br>0 = Non destructive sync is 1 CRE Clock width | |
| bit 9 | **DMC_DECB_CR4_ST_RA** | 1 = ramp simulation ON<br>0 = ramp simulation OFF | |
| bit 10 | **DMC_DECB_CR4_ST_FL** | 1 = Flasher is ON<br>0 = Flasher is OFF | |
| bit 11 | **DMC_DECB_CR4_ST_HE** | 1 = Heater is ON<br>0 = Heater is OFF | |
| bit 12-14 | **DMC_DECB_CR4_ST_SP2** | Spare | |
| bit 15 | **DMC_DECB_CR4_ST_CRPOW** | 1 = CRE powered on<br>0 = CRE powered off | |
| **326** | **DMC_DECB_BR_CM_4** | | **DEC** |
| | SCOS 2000 Display: | Decimal ( 0 = 0V, 4095 = +1V ) | |
| | Validity at startup: | Invalid | |

| | | | |
|---|---|---|---|
| | Validity during execution:<br>Available in:<br>Useful size (in bytes): | Valid only when Blue DEC is powered ON<br>All HK modes<br>2 (12bits) | |
| Description:<br>Bias R command readback | | | |
| **327** | **DMC_DECB_ZB_CM_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal ( 0 = 0V, 4095 = +1V )<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 (12bits) | |
| Description:<br>Zero Bias command readback | | | |
| **328** | **DMC_DECB_SR_RB_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>simulation register readback | | | |
| **329** | **DMC_DECB_TS_1_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 1 ohm)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Temperature Sensor 1 resistor value | | | |
| **330** | **DMC_DECB_TS_2_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 1 ohm)<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Temperature Sensor 2 resistor value | | | |
| **331** | **DMC_DECB_RO_CO_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>readout ID, counts from readouts_per_ramp-1 to 0 | | | |
| **332** | **DMC_DECB_RA_CO_4** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only when Blue DEC is powered ON<br>All HK modes<br>4 | |
| Description:<br>ramp counter, increments until reset | | | |
| **333** | **DMC_DECR_VDDD_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |

| | | | |
|---|---|---|---|
| Description:<br>VDDD Voltage Power Supply Group 1 | | | |
| **334** | **DMC_DECR_VSS_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VSS Voltage Power Supply Group 1 | | | |
| **335** | **DMC_DECR_VGND_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>GND Voltage Power Supply Group 1 | | | |
| **336** | **DMC_DECR_VCAN1_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N1 Voltage Power Supply Group 1 | | | |
| **337** | **DMC_DECR_VCAN2_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode N2 Voltage Power Supply Group 1 | | | |
| **338** | **DMC_DECR_V0BIAS1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Zero Bias Voltage Power Supply Group 1 | | | |
| **339** | **DMC_DECR_VBI_R_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Bias R Voltage Power Supply Group 1 | | | |
| **340** | **DMC_DECR_V0V_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>OV reference Voltage Power Supply Group 1 | | | |
| **341** | **DMC_DECR_VSCP_1** | | **DEC** |

| | | | |
|---|---|---|---|
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Cascode P Voltage Power Supply Group 1 | | | |
| **342** | **DMC_DECR_VDDR_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Current Mirror Voltage Power Supply Group 1 | | | |
| **343** | **DMC_DECR_VDDA_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VDDA Voltage Power Supply Group 1 | | | |
| **344** | **DMC_DECR_VWELL_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -6V, 65535 = 6V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Well Voltage Power Supply Group 1 | | | |
| **345** | **DMC_DECR_IDDA_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VDDA Current Power Supply Group 1 | | | |
| **346** | **DMC_DECR_IDDD_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VDDD Current Power Supply Group 1 | | | |
| **347** | **DMC_DECR_ISS_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VSS Current Power Supply Group 1 | | | |
| **348** | **DMC_DECR_IGND_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution: | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON | |

| | | | |
|---|---|---|---|
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>GND Current Power Supply Group 1 | | | |
| **349** | **DMC_DECR_HEAT_C** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -25mA, 32767 = 25mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Red DEC Heater Current | | | |
| **350** | **DMC_DECR_HEAT_V** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -37.5V, 32767 = 37.5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Red DEC Heater Voltage | | | |
| **351** | **DMC_DECR_REF_0V_1** | | **DEC** |
| | SCOS 2000 Display: | Decimal (-32767 = -5V, 32767 = 5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>0V Reference voltage for DEC Base Group 1 ADC | | | |
| **352** | **DMC_DECR_DCDC_T1** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 100Kohms) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DC/DC temperature. The temperature can be computed with:<br>T(K) = 1/(a0 + a1*ln(R) + a3*(ln(R)$^3$)<br>Where a0 = 1,2835e-3<br>a1 = 2,3646e-4<br>a3 = 9,1416e-8<br>Limit checking :<br>Warning when T(K) out of [243.15, 343.15]<br>Switch-off DMC when T(K) out of [218.15, 353.15] | | | |
| **353** | **DMC_DECR_SPARE5** | | **DEC** |
| | SCOS 2000 Display: | none | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>spare | | | |
| **354** | **DMC_DECR_DCDC_P5** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-1700mA, 32767=1700mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current from +5V power supply | | | |

| | | | |
|---|---|---|---|
| **355** | **DMC_DECR_AC_CUR** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-349.57mA, 32767=349.57mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>AC Current | | | |
| **356** | **DMC_DECR_TS_ST_1** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (4bits) | |
| Description:<br>Bit field showing the status of each of the detector array temperature sensors. 2bits for each sensors: | | | |
| 00 | Sensor inactive (measure is invalid) | | |
| 01 | Measure has been done using Low Gain Current Measure | | |
| 10 | Measure has been done using High Gain Current Measure | | |
| 11 | Error in measure (measure is invalid) | | |
| bits 0-1 | **DMC_DECR_TS_1_ST_1** | temperature sensor 1 status | |
| bits 2-3 | **DMC_DECR_TS_2_ST_1** | temperature sensor 2 status | |
| bits 4-15 | **DMC_DECR_TS_SP_1** | spare | |
| **357** | **DMC_DECR_CL_RO_1** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of CRE clocks per readout | | | |
| **358** | **DMC_DECR_RO_RA_1** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of readouts per ramp | | | |
| **359** | **DMC_DECR_CR_ST_1** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>CRE group 1 status word. Note, bit 0-14 are command readback. Bit 15 is the real status of the CRE power. | | | |
| bit 0 | **DMC_DECR_CR1_ST_POW** | 1 = CRE power on command readback<br>0 = CRE power off command readback | |
| bit 1 | **DMC_DECR_CR1_ST_SEL** | 1 = CRE Active<br>0 = CRE inactive | |
| bit 2-3 | **DMC_DECR_CR1_ST_CS** | Capacitor select read back<br>00 = 100fF<br>10 = 200fF<br>01 = 400fF | |

| | | | |
|---|---|---|---|
| | | 11 = 1pF | |
| bit 4 | **DMC_DECR_CR1_ST_CUR** | 1 = curing<br>0 = not curing | |
| bit 5 | **DMC_DECR_CR1_ST_SP1** | Spare | |
| bit 6 | **DMC_DECR_CR1_ST_SIM** | 1 = simulation mode<br>0 = nominal mode | |
| bit 7 | **DMC_DECR_CR1_ST_TE** | 1 = temperature sensors enabled<br>0 = temperature sensors disabled | |
| bit 8 | **DMC_DECR_CR1_ST_NDS** | 1 = Non destructive sync is 2 CRE Clock width<br>0 = Non destructive sync is 1 CRE Clock width | |
| bit 9 | **DMC_DECR_CR1_ST_RA** | 1 = ramp simulation ON<br>0 = ramp simulation OFF | |
| bit 10 | **DMC_DECB_CR1_ST_FL** | 1 = Flasher is ON<br>0 = Flasher is OFF | |
| bit 11 | **DMC_DECB_CR1_ST_HE** | 1 = Heater is ON<br>0 = Heater is OFF | |
| bit 12-14 | **DMC_DECR_CR1_ST_SP2** | Spare | |
| bit 15 | **DMC_DECR_CR1_ST_CRPO W** | 1 = CRE powered on<br><br>0 = CRE powered off | |
| **360** | **DMC_DECR_BR_CM_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal ( 0 = 0V, 4095 = +1V )<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 (12bits) | |
| Description:<br>Bias R command readback | | | |
| **361** | **DMC_DECR_ZB_CM_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal ( 0 = 0V, 4095 = +1V )<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 (12bits) | |
| Description:<br>Zero Bias command readback | | | |
| **362** | **DMC_DECR_SR_RB_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>simulation register readback | | | |
| **363** | **DMC_DECR_TS_1_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 1 ohm)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Temperature Sensor 1 resistor value | | | |
| **364** | **DMC_DECR_TS_2_1** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 1 ohm)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description: | | | |

Temperature Sensor 2 resistor value

| | | | |
|---|---|---|---|
| **365** | **DMC_DECR_RO_CO_1** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: readout ID, counts from readouts_per_ramp-1 to 0 | | | |
| **366** | **DMC_DECR_RA_CO_1** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: ramp counter, increments until reset | | | |
| **367** | **DMC_DECR_VDDD_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: VDD Voltage Power Supply Group 2 | | | |
| **368** | **DMC_DECR_VSS_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: VSS Voltage Power Supply Group 2 | | | |
| **369** | **DMC_DECR_VGND_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: GND Voltage Power Supply Group 2 | | | |
| **370** | **DMC_DECR_VCAN1_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Cascode N1 Voltage Power Supply Group 2 | | | |
| **371** | **DMC_DECR_VCAN2_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description: Cascode N2 Voltage Power Supply Group 2 | | | |
| **372** | **DMC_DECR_V0BIAS2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |

| | | | |
|---|---|---|---|
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Zero Bias Voltage Power Supply Group 2 | | | |
| **373** | **DMC_DECR_VBI_R_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Bias R Voltage Power Supply Group 2 | | | |
| **374** | **DMC_DECR_V0V_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>0V reference Voltage Power Supply Group 2 | | | |
| **375** | **DMC_DECR_VSCP_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Cascode P Voltage Power Supply Group 2 | | | |
| **376** | **DMC_DECR_VDDR_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current Mirror Voltage Power Supply Group 2 | | | |
| **377** | **DMC_DECR_VDDA_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDA Voltage Power Supply Group 2 | | | |
| **378** | **DMC_DECR_VWELL_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -6V, 65535 = 6V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Well Voltage Power Supply Group 2 | | | |
| **379** | **DMC_DECR_IDDA_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (0 = -0.6mA, 65535 = 0.6mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 2 | |
| Description:<br>VDDA Current Power Supply Group 2 | | | |
| **380** | **DMC_DECR_IDDD_2** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VDDD Current Power Supply Group 2 | | | |
| **381** | **DMC_DECR_ISS_2** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>VSS Current Power Supply Group 2 | | | |
| **382** | **DMC_DECR_IGND_2** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (0 = -0.6mA, 65535 = 0.6mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>GND Current Power Supply Group 2 | | | |
| **383** | **DMC_DECR_FLASH_C** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -25mA, 32767 = 25mA)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Red DEC Flasher Current | | | |
| **384** | **DMC_DECR_FLASH_V** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -37.5V, 32767 = 37.5V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>Red DEC Flasher Voltage | | | |
| **385** | **DMC_DECR_REF_0V2** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -5V, 32767 = 5V)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>0V Reference voltage for DEC Base Group 4 ADC | | | |
| **386** | **DMC_DECR_DCDC_T2** | | **DEC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (TBD)<br>Invalid<br>Valid only when Red DEC is powered ON<br>All HK modes<br>2 | |
| Description:<br>DC/DC temperature (not connected, spare sensor) | | | |

| | | | |
|---|---|---|---|
| **387** | **DMC_DECR_SPARE5B** | | **DEC** |
| | SCOS 2000 Display: | none | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>spare | | | |
| **388** | **DMC_DECR_DCDC_P15** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-144mA, 32767=144mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current from +15V power supply | | | |
| **389** | **DMC_DECR_DCDC_N15** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-144mA, 32767=144mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current from -15V power supply | | | |
| **390** | **DMC_DECR_TS_ST_2** | | **DEC** |
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (4bits) | |
| Description:<br>Bit field showing the status of each of the detector array temperature sensors. 2bits for each sensors: | | | |
| 00 | Sensor inactive (measure is invalid) | | |
| 01 | Measure has been done using Low Gain Current Measure | | |
| 10 | Measure has been done using High Gain Current Measure | | |
| 11 | Error in measure (measure is invalid) | | |
| bits 0-1 | **DMC_DECR_TS_1_ST_2** | temperature sensor 1 status | |
| bits 2-3 | **DMC_DECR_TS_2_ST_2** | temperature sensor 2 status | |
| bits 4-15 | **DMC_DECR_TS_SP_2** | spare | |
| **391** | **DMC_DECR_CL_RO_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of CRE clocks per readout | | | |
| **392** | **DMC_DECR_RO_RA_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Number of readouts per ramp | | | |

| 393 | DMC_DECR_CR_ST_2 | | DEC |
|---|---|---|---|
| | SCOS 2000 Display: | bit field | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
CRE group 2 status word. Note, bit 0-14 are command readback. Bit 15 is the real status of the CRE power.

| | | |
|---|---|---|
| bit 0 | **DMC_DECR_CR2_ST_POW** | 1 = CRE power on command readback<br>0 = CRE power off command readback |
| bit 1 | **DMC_DECR_CR2_ST_SEL** | 1 = CRE Active<br>0 = CRE inactive |
| bit 2-3 | **DMC_DECR_CR2_ST_CS** | Capacitor select read back<br>00 = 100fF<br>10 = 200fF<br>01 = 400fF<br>11 = 1pF |
| bit 4 | **DMC_DECR_CR2_ST_CUR** | 1 = curing<br>0 = not curing |
| bit 5 | **DMC_DECR_CR2_ST_SP1** | Spare |
| bit 6 | **DMC_DECR_CR2_ST_SIM** | 1 = simulation mode<br>0 = nominal mode |
| bit 7 | **DMC_DECR_CR2_ST_TE** | 1 = temperature sensors enabled<br>0 = temperature sensors disabled |
| bit 8 | **DMC_DECR_CR2_ST_NDS** | 1 = Non destructive sync is 2 CRE Clock width<br>0 = Non destructive sync is 1 CRE Clock width |
| bit 9 | **DMC_DECR_CR2_ST_RA** | 1 = ramp simulation ON<br>0 = ramp simulation OFF |
| bit 10 | **DMC_DECB_CR2_ST_FL** | 1 = Flasher is ON<br>0 = Flasher is OFF |
| bit 11 | **DMC_DECB_CR2_ST_HE** | 1 = Heater is ON<br>0 = Heater is OFF |
| bit 12-14 | **DMC_DECR_CR2_ST_SP2** | Spare |
| bit 15 | **DMC_DECR_CR2_ST_CRPOW** | 1 = CRE powered on<br>0 = CRE powered off |

| 394 | DMC_DECR_BR_CM_2 | | DEC |
|---|---|---|---|
| | SCOS 2000 Display: | Decimal ( 0 = 0V, 4095 = +1V ) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (12bits) | |

Description:
Bias R command readback

| 395 | DMC_DECR_ZB_CM_2 | | DEC |
|---|---|---|---|
| | SCOS 2000 Display: | Decimal ( 0 = 0V, 4095 = +1V ) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 (12bits) | |

Description:
Zero Bias command readback

| 396 | DMC_DECR_SR_RB_2 | | DEC |
|---|---|---|---|
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
simulation register readback

| | | | |
|---|---|---|---|
| **397** | **DMC_DECR_TS_1_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (1 unit = 1 ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Temperature Sensor 1 resistor value (not connected) | | | |
| **398** | **DMC_DECR_TS_2_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal (1 unit = 1 ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Temperature Sensor 2 resistor value (not connected) | | | |
| **399** | **DMC_DECR_RO_CO_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>readout ID, counts from readouts_per_ramp-1 to 0 | | | |
| **400** | **DMC_DECR_RA_CO_2** | | **DEC** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>ramp counter, increments until reset | | | |
| **401** | **DMC_SPARE4** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>spare | | | |
| **402** | **DMC_SPARE5** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>spare | | | |
| **403** | **DMC_SPARE6** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>spare | | | |
| **404** | **DMC_FPU_T_SENS_ST** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Bit field | |
| | Validity at startup: | Valid | |

| | | |
|---|---|---|
| | Validity during execution: | Always Valid |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 (14bits) |

Description:
Bit field showing the status of each of the FPU temperature sensors. 2bits for each sensors:

| | | |
|---|---|---|
| 00 | Sensor inactive (measure is invalid) | |
| 01 | Measure has been done using Low Gain Current Measure | |
| 10 | Measure has been done using High Gain Current Measure | |
| 11 | Error in measure (measure is invalid) | |
| bits 0-1 | **DMC_FPU_CH_TS_ST** | Chopper temperature sensor status |
| bits 2-3 | **DMC_FPU_CS_TS_ST** | CS temperature sensor status |
| bits 4-5 | **DMC_FPU_S1_TS_ST** | FPU temperature sensor 1 status |
| bits 6-7 | **DMC_FPU_S2_TS_ST** | FPU temperature sensor 2 status |
| bits 8-9 | **DMC_FPU_GR_TS_ST** | Grating temperature sensor status |
| bits 10-11 | **DMC_FPU_FWS_TS_ST** | FW Spec temperature sensor status |
| bits 12-13 | **DMC_FPU_FWP_TS_ST** | FW Photo temperature sensor status |
| bits 14-15 | **DMC_FPU_SPARE** | Spare |

| **405** | **DMC_FW_SPEC_TEMP** | **FPU_TEMP** |
|---|---|---|
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only when FPU T° measures enabled |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 |

Description:
FW SPEC Temperature sensor resistor value

| **406** | **DMC_FW_PHOT_TEMP** | **FPU_TEMP** |
|---|---|---|
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only when FPU T° measures enabled |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 |

Description:
FW PHOTO Temperature sensor resistor value

| **407** | **DMC_CHOPPER_TEMP** | **FPU_TEMP** |
|---|---|---|
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only when FPU T° measures enabled |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 |

Description:
Chopper Temperature sensor resistor value

| **408** | **DMC_GRATING_TEMP** | **FPU_TEMP** |
|---|---|---|
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only when FPU T° measures enabled |
| | Available in: | All HK modes |
| | Useful size (in bytes): | 2 |

Description:
Grating Temperature sensor resistor value

| **409** | **DMC_PSC_V1** | **HK** |
|---|---|---|
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-5.2A) |
| | Validity at startup: | Valid |

| | | | |
|---|---|---|---|
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply V1 current | | | |
| **410** | **DMC_PSC_V2** | | **HK** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-833mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply V2 current | | | |
| **411** | **DMC_PSC_V3** | | **HK** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-833mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply V3 current | | | |
| **412** | **DMC_PSC_V4** | | **HK** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-1.03A) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply V4 current | | | |
| **413** | **DMC_DCDC_TEMP** | | **HK** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 100Kohms) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Temperature of DMC DC/DC converters. The temperature can be computed with:<br>$T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3)$<br>Where a0 = 1,2835e-3<br>a1 = 2,3646e-4<br>a3 = 9,1416e-8<br>Limit checking :<br>Warning when T(K) out of [273.15, 343.15] (in raw values: [-287, -5350])<br>Switch-off DMC when T(K) > 353.15 (in raw values: <-206) | | | |
| **414** | **DMC_DSP_TEMP** | | **HK** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 200Kohms) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DMC DSP temperature. The temperature can be computed with:<br>$T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3)$<br>Where a0 = 8,79425e-4<br>a1 = 2,46538e-4<br>a3 = 1,16987e-7<br>Note that the maximum resistor value DMC can measure is 100kOhms<br>Limit checking :<br>Warning when T(K) out of [273.15, 353.15] (in raw values: [-363, -7270] | | | |

Switch-off DMC when T(K) > 373.15 (in raw values: <-205)

| | | | |
|---|---|---|---|
| **415** | **DMC_SPARE10** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
spare

| | | | |
|---|---|---|---|
| **416** | **DMC_SPARE11** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
spare

| | | | |
|---|---|---|---|
| **417** | **DMC_SPARE12** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
spare

| | | | |
|---|---|---|---|
| **418** | **DMC_SPARE13** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
spare

| | | | |
|---|---|---|---|
| **419** | **DMC_SPU_PSU_P15V** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-50V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU PSU +15V

| | | | |
|---|---|---|---|
| **420** | **DMC_SPU_SWL_TEMP** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 200Kohms) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU SWL DSP board temperature. The temperature can be computed with:
$T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3))$
Where a0 = 8,79425e-4
a1 = 2,46538e-4
a3 = 1,16987e-7

| | | | |
|---|---|---|---|
| **421** | **DMC_SPU_LWL_TEMP** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 200Kohms) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU LWL DSP board temperature. The temperature can be computed with:
$T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3))$
Where a0 = 8,79425e-4
a1 = 2,46538e-4
a3 = 1,16987e-7

| | | | |
|---|---|---|---|
| **422** | **DMC_SPU_PS_TEMP** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal (0=0ohms, -32767 = 200Kohms) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU Power supply temperature sensor. The temperature can be computed with:
$T(K) = 1/(a0 + a1*\ln(R) + a3*(\ln(R)^3))$
Where a0 = 8,79425e-4
a1 = 2,46538e-4
a3 = 1,16987e-7

| | | | |
|---|---|---|---|
| **423** | **DMC_SPU_VCC_CUR** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal(0 = 0mA, 32767 = 6.66A) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU 5V power supply current

| | | | |
|---|---|---|---|
| **424** | **DMC_SPU_VCC_VOL** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal(-32767 = -12.5V, 0 = 0V, 32767 = 12.5V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU 5V power supply voltage

| | | | |
|---|---|---|---|
| **425** | **DMC_SPU_VP_CUR** | | **SPU_HK** |
| | SCOS 2000 Display: | Decimal(-32767 = -180mA, 0 = 0mA, 32767 = 180mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
SPU 15V power supply current

| | | | |
|---|---|---|---|
| **426** | **DMC_FPU_T1_T** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when FPU T° measures enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
FPU Internal temperature sensor 1 resistor value

| | | | |
|---|---|---|---|
| **427** | **DMC_FPU_T2_T** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal (1 unit = 1ohm) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when FPU T° measures enabled | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 2 | |

Description:
FPU Internal temperature sensor 2 resistor value

| | | | |
|---|---|---|---|
| **428** | **DMC_REF_VOLT_0V** | | **HK** |

| | | | |
|---|---|---|---|
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -10V, 0 = 0V, 32767 = 10V)<br>Valid<br>Always Valid<br>All HK modes<br>2 | |
| Description:<br>Internal Reference voltage (0V) | | | |
| **429** | **DMC_CAL_SRC_TEMP** | | **FPU_TEMP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 1ohm)<br>Invalid<br>Valid only when FPU T° measures enabled<br>All HK modes<br>2 | |
| Description:<br>Calibration Source 1 housing temperature sensor resistor value. Note, if you are using redundant DMC, this value is the CS 2 temperature sensor resistor value | | | |
| **430** | **DMC_REF_VOLT_5V** | | **HK** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -10V, 0 = 0V, 32767 = 10V)<br>Valid<br>Always Valid<br>All HK modes<br>2 | |
| Description:<br>Internal Reference voltage (5V) | | | |
| **431** | **DMC_SPARE16** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | 0<br>Valid<br>Always Valid<br>All HK modes<br>2 | |
| Description:<br>spare | | | |
| **432** | **DMC_SPARE17** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | 0<br>Valid<br>Always Valid<br>All HK modes<br>2 | |
| Description:<br>spare | | | |
| **433** | **DMC_CUSTOM_ENT_1** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. By default, this entry is referencing the CRDCCP | | | |
| **434** | **DMC_CUSTOM_ENT_2** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |

| | | | |
|---|---|---|---|
| **435** | **DMC_CUSTOM_ENT_3** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **436** | **DMC_CUSTOM_ENT_4** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **437** | **DMC_CUSTOM_ENT_5** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **438** | **DMC_CUSTOM_ENT_6** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **439** | **DMC_CUSTOM_ENT_7** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **440** | **DMC_CUSTOM_ENT_8** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description: These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **441** | **DMC_CUSTOM_ENT_9** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | All HK modes | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 4 | |
| Description:<br>These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **442** | **DMC_CUSTOM_ENT10** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>These entries are customisable by configuring the Custom Hk Configuration Table. By this way, we can easily have additional Hk measures that were not foreseen when implementing the onboard software. | | | |
| **443** | **DMC_DET_SIM_STAT** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit Field (see description)<br>Valid<br>Always Valid<br>All HK modes<br>4 (25bits) | |
| Description:<br>Detector Simulator task status. | | | |
| bits 0-15 | **DMC_DSIM_ERROR** | Error code | |
| bit 16 | **DMC_DSIM_TASK_AL** | 1 = this task is running<br>0 = this task is not running | |
| bit 17 | **DMC_DSIM_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task | |
| bit 18 | **DMC_DSIM_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled | |
| bit 19 | **DMC_DSIM_SPARE1A** | Spare | |
| bit 20 | **DMC_DSIM_B_SIMUL** | 1 = Simulating Blue DEC<br>0 = Not simulating Blue DEC | |
| bit 21 | **DMC_DSIM_R_SIMUL** | 1 = Simulating Red DEC<br>0 = Not simulating Red DEC | |
| bit 22 | **DMC_DSIM_BOL_SIM** | 1 = Simulating BOLC<br>0 = Not simulating BOLC | |
| bit 23 | **DMC_DSIM_SPARE1B** | Spare | |
| bit 24 | **DMC_DSIM_TIME** | 1 = using simulated timing (the detector simulator is running and generating the timing)<br>0 = using real timing (replaces the science data received from DEC/BOLC by simulated readouts) | |
| bits 25-31 | **DMC_DSIM_SPARE7** | Spare | |
| **444** | **DMC_DET_SIM_PER** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (ms)<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>Detector simulator period | | | |
| **445** | **DMC_CS1_RES_VALUE** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution: | Decimal (1 unit = 100µohms)<br>Invalid<br>Valid only when CS1 is switched on | |

| | | | |
|---|---|---|---|
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>CS1 current resistor value | | | |
| **446** | **DMC_CS1_OUTPUT** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -10V, 32767 = 10V)<br>Invalid<br>Valid only when CS1 is switched on<br>All HK modes<br>2 | |
| Description:<br>CS1 controller commanded output voltage | | | |
| **447** | **DMC_CS2_RES_VALUE** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 100µohms)<br>Invalid<br>Valid only when CS2 is switched on<br>All HK modes<br>4 | |
| Description:<br>CS2 current resistor value | | | |
| **448** | **DMC_CS2_OUTPUT** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -10V, 32767 = 10V)<br>Valid<br>Valid only when CS2 is switched on<br>All HK modes<br>2 | |
| Description:<br>CS2 controller commanded output voltage | | | |
| **449** | **DMC_BOLC_STATUS** | | **BOLC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>All HK modes<br>2 | |
| Description:<br>BOLC Status word contained in the last packet received before HK sampling | | | |
| **450** | **DMC_BSPU_TR_MODE** | | **SPU** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>Blue Spu Transmission Mode | | | |
| **451** | **DMC_RSPU_TR_MODE** | | **SPU** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Valid<br>Always Valid<br>All HK modes<br>4 | |
| Description:<br>Red Spu Transmission Mode | | | |
| **452** | **DMC_GRAT_OUTPUT** | | **GRAT** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal<br>Invalid<br>Valid only while Grating controller is enabled<br>All HK modes<br>4 | |
| Description: | | | |

The commanded current output computed by the grating servo loop. (32767 = 555 mA and -32767 = -555 mA)

**453 DMC_OBT_COUNT** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Decimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 4 |

Description:
A counter which counts the OBT. This is the counter that is included in photometry packet header

**454 DMC_MIM_ST** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Hexadecimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 4 |

Description:
The MIM extension board status word. Bit field TBD.

**455 DMC_DM_SF_IND** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Decimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 1 |

Description:
DMC Data Memory Single Failure Index :
See section 'Detecting memory errors'

**456 DMC_PM_SF_IND** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Decimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 1 |

Description:
DMC Program Memory Single Failure Index :
See section 'Detecting memory errors'

**457 DMC_DM_DF_IND** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Decimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 1 |

Description:
DMC Data Memory Double Failure Index :
See section 'Detecting memory errors'

**458 DMC_PM_DF_IND** **OBSW**

| | |
|---|---|
| SCOS 2000 Display: | Decimal |
| Validity at startup: | Valid |
| Validity during execution: | Always Valid |
| Available in: | All HK modes |
| Useful size (in bytes): | 1 |

Description:
DMC Program Memory Double Failure Index :
See section 'Detecting memory errors'

**459 DMC_CS1_TARGET** **CS**

| | |
|---|---|
| SCOS 2000 Display: | Decimal (1 unit = 100µohms) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when CS1 is switched on |
| Available in: | All HK modes |

| | | |
|---|---|---|
| | Useful size (in bytes): | 4 |
| Description:<br>CS1 target resistor value | | |
| **460** | **DMC_CS2_TARGET** | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (1 unit = 100µohms)<br>Invalid<br>Valid only when CS2 is switched on<br>All HK modes<br>4 |
| Description:<br>CS2 target resistor value | | |
| **461** | **DMC_HK_CTRL_STAT** | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit Field (see description)<br>Valid<br>Always Valid<br>All HK modes<br>4 (19bits) |
| Description:<br>Nominal HK Controller task status. | | |
| bits 0-15 | **DMC_HKCO_ERROR** | Error code |
| bit 16 | **DMC_HKCO_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_HKCO_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_HKCO_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bits 19-31 | **DMC_HKCO_SPARE13** | Spare |
| **462** | **DMC_HK_DIAG_STAT** | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Bit Field (see description)<br>Valid<br>Always Valid<br>All HK modes<br>4 (21bits) |
| Description:<br>Diagnostic HK Controller task status. | | |
| bits 0-15 | **DMC_HKD_ERROR** | Error code |
| bit 16 | **DMC_HKD_TASK_AL** | 1 = this task is running<br>0 = this task is not running |
| bit 17 | **DMC_HKD_TASK_WR** | 1 = Any error occurred in the controller, the error code is copied in bits 0-15. The bit is cleared after each HK acquisition (unless bit 18 is set)<br>0 = No error in this task |
| bit 18 | **DMC_HKD_ERR_NS** | 1 = Error not signaled yet. (This bit is used internaly to make sure that all errors are signaled in HK packets at least once). If this bit is set, the error will also appear in the next HK packet<br>0 = No error waiting to be signaled |
| bit 19 | **DMC_HKD_SPARE1** | Spare |
| bit 20 | **DMC_HKD_DIAGMODE** | 1 = Currently acquiring diagnostic HK<br>0 = no diagnostic HK acquired now |
| bits 21-31 | **DMC_HKD_SPARE11** | Spare |
| **463** | **DMC_HK_DIAG_PERI** | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution: | Decimal (units = ms)<br>Valid<br>Always Valid |

| | | | |
|---|---|---|---|
| | Available in: | All HK modes | |
| | Useful size (in bytes): | 4 | |
| Description:<br>Period of acquisition of diagnostic housekeeping. When diagnostic hk is synchronized on a detector, the period is 0. | | | |
| **464** | **DMC_LAST_ERR_ID** | | **OBSW** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 1 (4bits) | |
| Description:<br>Last Error Buffer Index : Indicates the current position in the Last Errors Buffer. Index is 0 based and indicates the next position to be filled. (Note: index 0 is DMC_LAST_ER_BF1) | | | |
| **465** | **DMC_LAST_ER_BF1** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **466** | **DMC_LAST_ER_BF2** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **467** | **DMC_LAST_ER_BF3** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **468** | **DMC_LAST_ER_BF4** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **469** | **DMC_LAST_ER_BF5** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |

| | | | |
|---|---|---|---|
| **470** | **DMC_LAST_ER_BF6** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **471** | **DMC_LAST_ER_BF7** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **472** | **DMC_LAST_ER_BF8** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **473** | **DMC_LAST_ER_BF9** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **474** | **DMC_LAST_ER_BF10** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **475** | **DMC_LAST_ER_BF11** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each).

| | | | |
|---|---|---|---|
| **476** | **DMC_LAST_ER_BF12** | | **OBSW** |
| | SCOS 2000 Display: | Hexadecimal | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Nominal HK Only | |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **477** | **DMC_LAST_ER_BF13** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>Nominal HK Only<br>2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **478** | **DMC_LAST_ER_BF14** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>Nominal HK Only<br>2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **479** | **DMC_LAST_ER_BF15** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>Nominal HK Only<br>2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **480** | **DMC_LAST_ER_BF16** | | **OBSW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Hexadecimal<br>Valid<br>Always Valid<br>Nominal HK Only<br>2 | |
| Description:<br>Last Error Buffer : A 16 words circular buffer containing the last 16 errors generated by all the tasks. Each word contains 1 error codes (16 bits each). | | | |
| **481** | **BOLC_HK_197** | | **BOLC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br><br>Available in:<br>Useful size (in bytes): | DISPLAY_HEX<br>AVM : Valid, Further models : invalid<br>AVM : Always valid. Further models : Valid only when BOLC is ON and the connection between DMC and BOLC is established<br>All HK modes<br>4 | |
| Description:<br>BOLC HK entry 197 | | | |

…

| | | | |
|---|---|---|---|
| **508** | **BOLC_HK_224** | | **BOLC** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br><br>Available in:<br>Useful size (in bytes): | DISPLAY_HEX<br>AVM : Valid, Further models : invalid<br>AVM : Always valid. Further models : Valid only when BOLC is ON and the connection between DMC and BOLC is established<br>All HK modes<br>4 | |
| Description:<br>BOLC HK entry 224 | | | |

### 5.3.3 List of Diagnostic housekeeping Measure

| | | | |
|---|---|---|---|
| **512** | **DMC_GR_IND_READ** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 4 | |
| Description: The position as read by the inductosyn (before all transformation and correction by software) | | | |
| **513** | **DMC_GR_TURN_CAR** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 4 | |
| Description: Inductosyn big jumps (on complete turn) carry counter | | | |
| **514** | **DMC_GR_PER_CAR** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 4 | |
| Description: Inductosyn small jumps (one period) carry counter | | | |
| **515** | **DMC_GR_DEG_POS** | | **GRAT** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while Grating is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 4 | |
| Description: Grating absolute position in degraded mode. Note: this variable is updated only when entering the degraded mode (if grating is switched on) and during a degraded move. Unit = index in the sine table (0 - 16383), 1 unit = 13.18 arcsec | | | |
| **516** | **DMC_SPARE_DIAG7** | | **SPARE** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **517** | **DMC_SPARE_DIAG8** | | **SPARE** |
| | SCOS 2000 Display: | Decimal | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **518** | **DMC_SPARE_DIAG1** | | **SPARE** |
| | SCOS 2000 Display: | Decimal | |

| | | |
|---|---|---|
| | Validity at startup: | Invalid |
| | Validity during execution: | Always Invalid |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>spare | | |
| **519** | **DMC_SPARE_DIAG2** | **SPARE** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Invalid |
| | Validity during execution: | Always Invalid |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>spare | | |
| **520** | **DMC_SPARE_DIAG3** | **SPARE** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Invalid |
| | Validity during execution: | Always Invalid |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>spare | | |
| **521** | **DMC_SPARE_DIAG4** | **SPARE** |
| | SCOS 2000 Display: | Decimal |
| | Validity at startup: | Invalid |
| | Validity during execution: | Always Invalid |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>spare | | |
| **522** | **DMC_CS1_VOLT_0V** | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -6.25V, 32767 = 6.25V) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only while CS1 is powered on |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>0V reference | | |
| **523** | **DMC_CS1_VOLT_N5V** | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -6.25V, 32767 = 6.25V) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only while CS1 is powered on |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>-4V reference | | |
| **524** | **DMC_CS1_VOLT_P5V** | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -6.25V, 32767 = 6.25V) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only while CS1 is powered on |
| | Available in: | Diag HK Only |
| | Useful size (in bytes): | 2 |
| Description:<br>+4V reference | | |
| **525** | **DMC_CS1_VOLT_DAC_OUT** | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -12.5V, 32767 = 12.5V) |
| | Validity at startup: | Invalid |
| | Validity during execution: | Valid only while CS1 is powered on |
| | Available in: | Diag HK Only |

| | | | |
|---|---|---|---|
| | Useful size (in bytes): | 2 | |
| Description:<br>Calibration source DAC output = measured real output voltage to calibration source. | | | |
| **526** | **DMC_CS1_VOLT_SG** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -2.5V, 32767 = 2.5V)<br>Invalid<br>Valid only while CS1 is powered on<br>Diag HK Only<br>2 | |
| Description:<br>Small Gain sensor voltage | | | |
| **527** | **DMC_CS1_VOLT_BG** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -25mV, 32767 = 25mV)<br>Invalid<br>Valid only while CS1 is powered on<br>Diag HK Only<br>2 | |
| Description:<br>Big Gain sensor voltage | | | |
| **528** | **DMC_CS1_CUR_SG** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -25mA, 32767 = 25mA)<br>Invalid<br>Valid only while CS1 is powered on<br>Diag HK Only<br>2 | |
| Description:<br>Small Gain sensor current. Note, this value is coming from the voltage measured on a 100ohm reference resistor. If this resistor is not exactly 100 ohms, the conversion must be adjusted | | | |
| **529** | **DMC_CS1_CUR_BG** | | **CS** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (-32767 = -250µA, 32767 = 250µA)<br>Invalid<br>Valid only while CS1 is powered on<br>Diag HK Only<br>2 | |
| Description:<br>Big Gain sensor current. Note, this value is coming from the voltage measured on a 100ohm reference resistor. If this resistor is not exactly 100 ohms, the conversion must be adjusted | | | |
| **530** | **DMC_CS1_SPARE1** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | 0<br>Invalid<br>Always Invalid<br>Diag HK Only<br>2 | |
| Description:<br>spare | | | |
| **531** | **DMC_CS1_SPARE2** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | 0<br>Invalid<br>Always Invalid<br>Diag HK Only<br>2 | |
| Description:<br>spare | | | |
| **532** | **DMC_CS1_SPARE3** | | **SPARE** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | 0<br>Invalid<br>Always Invalid<br>Diag HK Only<br>2 | |

Description:
spare

| | | | |
|---|---|---|---|
| 533 | DMC_CS1_SPARE4 | | SPARE |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| 534 | DMC_CS1_SPARE5 | | SPARE |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| 535 | DMC_CS1_SPARE6 | | SPARE |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| 536 | DMC_CS1_SPARE7 | | SPARE |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| 537 | DMC_CS1_SPARE8 | | SPARE |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| 538 | DMC_CS2_VOLT_0V | | CS |
| | SCOS 2000 Display: | Decimal (-32767 = -6.25V, 32767 = 6.25V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: 0V reference | | | |
| 539 | DMC_GR_IND_SINE | | GRAT |
| | SCOS 2000 Display: | Decimal (-32767 = -5V, 32767 = 5V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: Grating inductosyn sine amplitude (zero to peak)<br>Note : this is not directly a measurement of the amplitude, conversion using calibration is needed. | | | |

| | | | |
|---|---|---|---|
| **540** | **DMC_GR_IND_COS** | | **GRAT** |
| | SCOS 2000 Display: | Decimal (-32767 = -5V, 32767 = 5V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Grating inductosyn cos amplitude (zero to peak)<br>Note : this is not directly a measurement of the amplitude, conversion using calibration is needed. | | | |
| **541** | **DMC_CS2_VOLT_DAC_OUT** | | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -12.5V, 32767 = 12.5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Calibration source DAC output = measured real output voltage to calibration source. | | | |
| **542** | **DMC_CS2_VOLT_SG** | | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -2.5V, 32767 = 2.5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Small Gain sensor voltage | | | |
| **543** | **DMC_CS2_VOLT_BG** | | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -25mV, 32767 = 25mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Big Gain sensor voltage | | | |
| **544** | **DMC_CS2_CUR_SG** | | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -25mA, 32767 = 25mA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Small Gain sensor current. Note, this value is coming from the voltage measured on a 100ohm reference resistor. If this resistor is not exactly 100 ohms, the conversion must be adjusted | | | |
| **545** | **DMC_CS2_CUR_BG** | | **CS** |
| | SCOS 2000 Display: | Decimal (-32767 = -250µA, 32767 = 250µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only while CS2 is powered on | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Big Gain sensor current. Note, this value is coming from the voltage measured on a 100ohm reference resistor. If this resistor is not exactly 100 ohms, the conversion must be adjusted | | | |
| **546** | **DMC_GR_LL1_CUR** | | **SPARE** |
| | SCOS 2000 Display: | Decimal (+/-32767 = +/-502mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Grating Launch Lock Motor 1 Current | | | |

| | | | |
|---|---|---|---|
| **547** | **DMC_CS2_SPARE2** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **548** | **DMC_CS2_SPARE3** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **549** | **DMC_CS2_SPARE4** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **550** | **DMC_CS2_SPARE5** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **551** | **DMC_CS2_SPARE6** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **552** | **DMC_CS2_SPARE7** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **553** | **DMC_CS2_SPARE8** | | **SPARE** |
| | SCOS 2000 Display: | 0 | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Always Invalid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description: spare | | | |
| **554** | **DMC_PSU_5V_VOLT** | | **HK** |
| | SCOS 2000 Display: | Decimal(-32767 = -10V, 0 = 0V, 32767 = 10V) | |
| | Validity at startup: | Valid | |

| | | | |
|---|---|---|---|
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply Voltage (+5V) | | | |
| **555** | **DMC_FWSPEC_POS_A** | | **FW** |
| | SCOS 2000 Display: | Decimal(-32767 = -102mV, 0 = 0mV, 32767 = 102mV) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>FW Spec Position sensor A | | | |
| **556** | **DMC_FW_GR_VMOTA** | | **FW** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-30V for FM and +/- 10V for QM) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>FW-Grat Amplifier voltage phase A | | | |
| **557** | **DMC_CHOP_VA** | | **CHOP** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-19.2V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Chopper amplifier voltage side A | | | |
| **558** | **DMC_PSU_P15V_V** | | **HK** |
| | SCOS 2000 Display: | Decimal(-32767 = -20V, 0 = 0V, 32767 = 20V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Power Supply Voltage (+15V) | | | |
| **559** | **DMC_FWSPEC_POS_B** | | **FW** |
| | SCOS 2000 Display: | Decimal(-32767 = -102mV, 0 = 0mV, 32767 = 102mV) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>FW Spec Position sensor B | | | |
| **560** | **DMC_FW_GR_IMOTA** | | **GRAT_FW** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-554mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>FW-Grat Amplifier current phase A (note: not read on EM hardware) | | | |
| **561** | **DMC_CHOP_IA** | | **CHOP** |
| | SCOS 2000 Display: | Decimal(+/-32767 = -147mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |

| | | | |
|---|---|---|---|
| Description:<br>Chopper amplifier current side A (note: not read on EM hardware) | | | |
| **562** | **DMC_PSU_N15V_V** | | **HK** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -20V, 0 = 0V, 32767 = 20V)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>Power Supply Voltage (-15V) | | | |
| **563** | **DMC_FWPHOT_POS_A** | | **FW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -102mV, 0 = 0mV, 32767 = 102mV)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>FW Photo Position sensor A | | | |
| **564** | **DMC_FW_GR_VMOTB** | | **GRAT_FW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(+/-32767 = +/-30V for FM and +/- 10V for QM)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>FW-Grat Amplifier voltage phase B | | | |
| **565** | **DMC_CHOP_VB** | | **CHOP** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(+/-32767 = -19.2V)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>Chopper amplifier voltage side B | | | |
| **566** | **DMC_ADC_VOLT** | | **HK** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -10V, 0 = 0V, 32767 = 10V)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>ADC Voltage (+2.5V) | | | |
| **567** | **DMC_FW_GR_IMOTB** | | **GRAT_FW** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(+/-32767 = +/-554mA)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>FW-Grat amplifier current phase B (note: not read on EM hardware) | | | |
| **568** | **DMC_PSU_P28V_V** | | **HK** |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal(-32767 = -20V, 0 = 0V, 32767 = 20V)<br>Valid<br>Always Valid<br>Diag HK Only<br>2 | |
| Description:<br>Power Supply Voltage (+28V). Note, the 28V is actually made of a -14V and a +14V. The +14V is measured here. | | | |

| | | | |
|---|---|---|---|
| **569** | **DMC_FWPHOT_POS_B** | | **FW** |
| | SCOS 2000 Display: | Decimal(-32767 = -102mV, 0 = 0mV, 32767 = 102mV) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>FW Photo Position sensor B | | | |
| **570** | **DMC_GR_LL2_CUR** | | **GRAT** |
| | SCOS 2000 Display: | Decimal (+/-32767 = +/-502mA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Grating Launch Lock Motor 2 Current | | | |
| **571** | **DMC_T_SE_SRC1_LG** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(-32767 = -113µA, 0 = 0µA, 32767 = 113µA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current through the first chain of FPU T° sensors (low gain) | | | |
| **572** | **DMC_T_SE_SRC1_HG** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(-32767 = -4.58µA, 0 = 0µA, 32767 = 4.58µA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current through the first chain of FPU T° sensors (high gain) | | | |
| **573** | **DMC_T_SE_SRC1_V1** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(-32767 = -10V, 0 = 0V, 32767 = 10V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage applied to the first chain of FPU T° sensors (negative) | | | |
| **574** | **DMC_T_SE_SRC1_V2** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(+/-32767 = +/-9.97V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage applied to the second chain of FPU T° sensors (positive) | | | |
| **575** | **DMC_T_SE_SRC2_LG** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(-32767 = -113µA, 0 = 0µA, 32767 = 113µA) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current through the second chain of FPU T° sensors (low gain) | | | |
| **576** | **DMC_T_SE_SRC2_HG** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(-32767 = -4.58µA, 0 = 0µA, 32767 = 4.58µA) | |
| | Validity at startup: | Valid | |

| | | | |
|---|---|---|---|
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current through the second chain of FPU T° sensors (high gain) | | | |
| **577** | **DMC_T_SE_SRC2_V1** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(+/-32767 = -9.97V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage applied to the second chain of FPU T° sensors (negative) | | | |
| **578** | **DMC_T_SE_SRC2_V2** | | **FPU_TEMP** |
| | SCOS 2000 Display: | Decimal(+/-32767 = -9.97V) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Always Valid | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage applied to the first chain of FPU T° sensors (positive) | | | |
| **579** | **DMC_DB_TS12CBS_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (32767 = -2.222µA, 32767 = 2.222µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current in temperature sensors 1 and 2 Power Supply Group 3. Same measure as next one but with a bigger scale. | | | |
| **580** | **DMC_DB_TS12CSS_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767 = -55.555µA, 32767 = 55.555µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current in temperature sensors 1 and 2 Power Supply Group 3. Same measure as previous one but with a smaller scale. If value is out of [-2µA, 2µA], use the other value. | | | |
| **581** | **DMC_DECB_TS1_V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 1 Power Supply Group 3 | | | |
| **582** | **DMC_DECB_TS2_V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 2 Power Supply Group 3 | | | |
| **583** | **DMC_DECB_PS_GEN3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |

| | | | |
|---|---|---|---|
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (+) Power Supply Group 3 | | | |
| **584** | **DMC_DECB_NS_GEN3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (-) Power Supply Group 3 | | | |
| **585** | **DMC_DECB_D5V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Digital +5V | | | |
| **586** | **DMC_DECB_D2_5V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Digital +2.5V | | | |
| **587** | **DMC_DECB_A5V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Analog +5V | | | |
| **588** | **DMC_DECB_R5V_3** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Reference +5V | | | |
| **589** | **DMC_DB_TS12CBS_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (32767 = -2.222µA, 32767 = 2.222µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current in temperature sensors 1 and 2 Power Supply Group 4. Same measure as next one but with a bigger scale. | | | |
| **590** | **DMC_DB_TS12CSS_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767 = -55.555µA, 32767 = 55.555µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |

Description:
Current in temperature sensors 1 and 2 Power Supply Group 4. Same measure as previous one but with a smaller scale. If value is out of [-2µA, 2µA], use the other value.

| | | | |
|---|---|---|---|
| **591** | **DMC_DECB_TS1_V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 1 Power Supply Group 4 | | | |
| **592** | **DMC_DECB_TS2_V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 2 Power Supply Group 4 | | | |
| **593** | **DMC_DECB_PS_GEN4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (+) Power Supply Group 4 | | | |
| **594** | **DMC_DECB_NS_GEN4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (-) Power Supply Group 4 | | | |
| **595** | **DMC_DB_DC_P15V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-25V, 32767=+25V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DC/DC +15V | | | |
| **596** | **DMC_DB_DC_N15V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-25V, 32767=+25V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DC/DC -15V | | | |
| **597** | **DMC_DECB_A5V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Analog +5V | | | |

| | | | |
|---|---|---|---|
| **598** | **DMC_DECB_R5V_4** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Blue DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Reference +5V | | | |
| **599** | **DMC_DR_TS12CBS_1** | | **DEC** |
| | SCOS 2000 Display: | Dec (32767 = -2.222µA, 32767 = 2.222µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current in temperature sensors 1 and 2 Power Supply Group 1. Same measure as next one but with a bigger scale. | | | |
| **600** | **DMC_DR_TS12CSS_1** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767 = -55.555µA, 32767 = 55.555µA) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Current in temperature sensors 1 and 2 Power Supply Group 1. Same measure as previous one but with a smaller scale. If value is out of [-2µA, 2µA], use the other value. | | | |
| **601** | **DMC_DECR_TS1_V_1** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 1 Power Supply Group 1 | | | |
| **602** | **DMC_DECR_TS2_V_1** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Nominal HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 2 Power Supply Group 1 | | | |
| **603** | **DMC_DECR_PS_GEN1** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (+) Power Supply Group 1 | | | |
| **604** | **DMC_DECR_NS_GEN1** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (-) Power Supply Group 1 | | | |
| **605** | **DMC_DECR_D5V_1** | | **DEC** |

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Digital +5V

**606 DMC_DECR_D2_5V_1 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Digital +2.5V

**607 DMC_DECR_A5V_1 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Analog +5V

**608 DMC_DECR_R5V_1 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Reference +5V

**609 DMC_DR_TS12CBS_2 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (32767 = -2.222µA, 32767 = 2.222µA) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Current in temperature sensors 1 and 2 Power Supply Group 2. Same measure as next one but with a bigger scale.

**610 DMC_DR_TS12CSS_2 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767 = -55.555µA, 32767 = 55.555µA) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Current in temperature sensors 1 and 2 Power Supply Group 2. Same measure as previous one but with a smaller scale. If value is out of [-2µA, 2µA], use the other value.

**611 DMC_DECR_TS1_V_2 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) |
| Validity at startup: | Invalid |
| Validity during execution: | Valid only when Red DEC is powered ON |
| Available in: | Diag HK Only |
| Useful size (in bytes): | 2 |

Description:
Voltage in temperature sensor 1 Power Supply Group 2

**612 DMC_DECR_TS2_V_2 DEC**

| | |
|---|---|
| SCOS 2000 Display: | Dec (-32767= -7.143mV, 32767 = 7.143mV) |

| | | | |
|---|---|---|---|
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Voltage in temperature sensor 2 Power Supply Group 2 | | | |
| **613** | **DMC_DECR_PS_GEN2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (+) Power Supply Group 2 | | | |
| **614** | **DMC_DECR_NS_GEN2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-5V, 32767=+5V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Sensor Generator (-) Power Supply Group 2 | | | |
| **615** | **DMC_DR_DCDC_P15V_2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-25V, 32767=+25V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DC/DC +15V | | | |
| **616** | **DMC_DR_DCDC_N15V_2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-25V, 32767=+25V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>DC/DC -15V | | | |
| **617** | **DMC_DECR_A5V_2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Analog +5V | | | |
| **618** | **DMC_DECR_R5V_2** | | **DEC** |
| | SCOS 2000 Display: | Dec (-32767=-10V, 32767=+10V) | |
| | Validity at startup: | Invalid | |
| | Validity during execution: | Valid only when Red DEC is powered ON | |
| | Available in: | Diag HK Only | |
| | Useful size (in bytes): | 2 | |
| Description:<br>Reference +5V | | | |
| **619** | **DMC_TS_FW_SPEC_V** | | |
| | SCOS 2000 Display: | Decimal (+/-32767=-14,54mV) | |
| | Validity at startup: | Valid | |
| | Validity during execution: | Spare | |
| | Available in: | Diag HK Only | |

| | | |
|---|---|---|
| | Useful size (in bytes): | 2 |
| Description:<br>FW SPEC temperature sensor voltage | | |
| **620** | **DMC_TS_FW_PHOT_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,45mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>FW PHOTO temperature sensor voltage | | |
| **621** | **DMC_TS_GRAT_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,50mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>Grating temperature sensor voltage | | |
| **622** | **DMC_TS_CHOP_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,51mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>Chopper temperature sensor voltage | | |
| **623** | **DMC_TS_FPU_T1_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,50mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>FPU T1 temperature sensor voltage | | |
| **624** | **DMC_TS_FPU_T2_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,55mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>FPU T2 temperature sensor voltage | | |
| **625** | **DMC_TS_BB_V** | |
| | SCOS 2000 Display:<br>Validity at startup:<br>Validity during execution:<br>Available in:<br>Useful size (in bytes): | Decimal (+/-32767=-14,50mV)<br>Valid<br>Spare<br>Diag HK Only<br>2 |
| Description:<br>BB temperature sensor voltage | | |

## 5.4 Error Management

When an error occurs in any of the tasks of the DMC OBS, the following actions are performed:

- The error code is stored in the 16lsb of the task status word.
- The error code is stored in the 16lsb of the DMC_SW_GLOBAL_ST and the bit17 is raised.
- The error code is stored in the DMC_LAST_ER_BF array at the position given by DMC_LAST_ERR_ID
- DMC_LAST_ERR_ID is incremented.

The error codes are stored in the tasks status word and in DMC_SW_GLOBAL_ST only during one HK packet (unless if the bit 18 is set, in this case, the error code can stay during 2 HK packets).

The DMC_LAST_ER_BF is a circular buffer that contains the 16 last errors that have been signaled in DMC_SW_GLOBAL_ST. The current position in the circular buffer is given by DMC_LAST_ERR_ID that contains the next position to be filled.

### 5.4.1 Error code table

The table below presents all the error codes that may appear in the 16 LSB of the task status.

Note :

- Error codes (and the "Task in error" bit in the status of the task) are reset after each nominal housekeeping acquisition.
- The error codes are available in "error.h".
- We make a difference between
  - general errors that can be caused by an external process (most of the errors); i.e. device not working properly, received a bad packet, … General errors code have the form 0x??0?
  - internal errors are caused by the application itself (shall not appear once development is finished but, who knows …); i.e internal bug, lack of resource, … Internal errors code have the form 0x??1?

| Error Code | Comment |
|---|---|
| | |
| 0x0200 | **ERR_LINK_SPU_BLUE**<br>Any error occurred in relation with the spacewire link to the Blue SPU |
| 0x0201 | **ERR_LINK_SPU_RED**<br>Any error occurred in relation with the spacewire link to the Red SPU |
| 0x0202 | **ERR_LINK_ DPU** |

| | |
|---|---|
| | Any error occurred in relation with the spacewire link to the DPU |
| 0x0203 | **ERR_LINK_BOL**<br>Any error occurred in relation with the spacewire link to the BOL |
| 0x0204 | **ERR_LINK_DEC_BLUE**<br>Any error occurred in relation with the spacewire link to the Blue DEC |
| 0x0205 | **ERR_LINK_DEC_RED**<br>Any error occurred in relation with the spacewire link to the Red DEC |
| 0x0210 | **ERR_SMCS_DRIVER_COULD_NOT_CREATE_TIMER**<br>Internal error : Could not create the timer for the SMCS driver. |
| | |
| 0x0300 | **ERR_SEQUENCER_UNKNOWN_COMMAND**<br>Sequencer tries to execute a command (trigger command or inside a sequence) with an invalid ID. |
| 0x0301 | **ERR_SEQUENCER_INVALID_PARAMETERS**<br>Sequencer tries to execute a command (trigger command or inside a sequence) with invalid parameters. |
| 0x0302 | **ERR_SEQUENCER_COMMAND_NOT_AVAILABLE_IN_THIS_MODE**<br>Sequencer tries to execute a trigger command while a sequence is being executed (and this command can not be executed during the execution of a sequence) or, inside a sequence, it tries to execute a command that is only available as a trigger command. |
| 0x0303 | **ERR_SEQUENCER_SYNC_TIME_OUT**<br>Sequencer has been waiting too long for the synchronisation signal (timing signal issues when a ramp is finished (in spectroscopy) or when a readout arrives (in photometry). |
| 0x0304 | **ERR_SEQUENCER_TOO_MANY_NESTED_LOOPS**<br>Too many nested loops in a sequence (maximum is 5) |
| 0x0305 | **ERR_SEQUENCER_LOOP_END_OF_LOOP_MISMATCH**<br>There are more END_OF_LOOP as LOOP commands in the sequence. |
| 0x0306 | **ERR_SEQUENCER_RELATIVE_SETTING_OUT_OF_RANGE**<br>The sequencer tries to execute a command (trigger command or inside a sequence) with relative parameters (i.e. MOVE_GRATING_RELATIVE). The relative parameter sets the absolute parameter out of the accepted range. |
| 0x0307 | **ERR_SEQUENCER_COULD_NOT_EXECUTE_COMMAND**<br>An error occurred while executing the command. Or, it was not possible to start the execution of this command in the current status of the OBS. |
| 0x0310 | **ERR_SEQUENCER_UNKNOWN_MESSAGE_TYPE**<br>Internal error : Sequencer received an unknown message type on its FIFO. |
| | |

| | |
|---|---|
| 0x0400 | **ERR_DPU_RECEIVER_UNKNOWN_MSG_TYPE**<br>The DPU receiver has received a unknown message format. Note that a NACK will be generated also. |
| 0x0401 | **ERR_DPU_RECEIVER_INVALID_ADDRESS**<br>Invalid address in a memory load/dump/check command<br>Note that a NACK will be generated also. |
| 0x0402 | **ERR_DPU_RECEIVER_INVALID_PARAM_ID**<br>Invalid parameter ID in a write command.<br>Note that a NACK will be generated also. |
| 0x0403 | **ERR_DPU_RECEIVER_INVALID_LENGTH**<br>Invalid length in a memory load/dump/check/write command.<br>Note that a NACK will be generated also. |
| 0x0404 | **ERR_DPU_RECEIVER_INVALID_MEM_ID**<br>Received a load command trying to access memory outside DRAM<br>Note that a NACK will be generated also. |
| 0x0410 | **ERR_DPU_RECEIVER_TIME_OUT_ON_DUMP_BUFFER**<br>Internal Error : time-out while waiting that the DPU sender sends the last Dump packet |
| | |
| 0x0510 | **ERR_DPU_SENDER_UNKNOWN_FIFO_MSG**<br>Internal error : DPU Sender received an unknown message type on its FIFO. |
| 0x0511 | **ERR_DPU_SENDER_UNRECOGNISED_ERROR_CODE**<br>Internal error : DPU Sender received an unknown error code for the type of message it is supposed to send. |
| | |
| 0x0600 | **ERR_DEC_RECEIVER_INVALID_READOUT_INTERVAL**<br>The Dec Receiver has received a readout at a rate that was not expected. |
| 0x0601 | **ERR_DEC_RECEIVER_INVALID_READOUT_COUNTER**<br>The readout counter received from DEC is bigger than the number of readout in a ramp. |
| | |
| 0x0710 | **ERR_DET_SIMULATOR_COULD_NOT_CREATE_TIMER**<br>Internal error : could not create the timer for the detector simulator |
| | |
| 0x0800 | **ERR_HK_INVALID_MEASURE_ID**<br>The diagnostic HK list (or very improbably the nominal HK list) contained an invalid measure ID. |
| 0x0801 | **ERR_HK_MEASURE_NOT_AVAILABLE_IN_DIAG_MODE** |

| | |
|---|---|
| | The diagnostic HK list contains the ID of a measure that is not available in diagnostic mode.<br>Note that, in this case, the acquisition will be performed anyway but the measure will not be included at all in the Hk diagnostic packet.<br>Note also that, the error will be generated at each diagnostic acquisition (it may be 256 times a second). So, this will probably fill the Last Error Buffer very fast. |
| 0x0810 | **ERR_HK_COULD_NOT_CREATE_HK_TIMER**<br>Internal error : Could not create the timer for the nominal HK. |
| 0x0811 | **ERR_HK_DIAG_COULD_NOT_CREATE_HK_DIAG_TIMER**<br>Internal error : Could not create the timer for the diagnostic HK. |
| | |
| 0x0900 | **ERR_DEC_CONTROLLER_LINK_NOT_CONNECTED**<br>Unable to send a command to DEC since its link is not connected |
| 0x0901 | **ERR_DEC_CONTROLLER_NO_RAMP_AFTER_RESET**<br>After a reset (write parameters), no ramp has been received (time-out reached) |
| 0x0902 | **ERR_DEC_CONTROLLER_OTHER_CRE_ON**<br>Trying to switch-on a DEC while the other CREs are already powered ON |
| 0x0910 | **ERR_DEC_CONTROLLER_UNKNOWN_FIFO_MSG**<br>Internal error : DEC Controller received an unknown message type on its FIFO. |
| | |
| 0x0A00 | **ERR_BOL_CONTROLLER_LINK_NOT_CONNECTED**<br>Unable to send a command to DEC since its link is not connected |
| 0x0A10 | **ERR_BOL_CONTROLLER_UNKNOWN_FIFO_MSG**<br>Internal error : DEC Controller received an unknown message type on its FIFO. |
| | |
| 0x0B13 | **ERR_CHOPPER_CONTROLLER_FOLLOWING_ERROR**<br>The current chopper error is bigger than the error limit |
| 0x0B15 | **ERR_CHOPPER_CONTROLLER_POSITION_ERROR**<br>The current chopper position is bigger than the position limit |
| 0x0B23 | **ERR_GRATING_CONTROLLER_FOLLOWING_ERROR**<br>The current grating error is bigger than the error limit |
| 0x0B24 | **ERR_GRATING_CONTROLLER_POWER_LIMIT_ERROR**<br>The grating output was equal to the output limit during 5 seconds |
| | |
| 0x0C00 | **ERR_PACKET_ENCODER_INVALID_READOUT_COUNTER**<br>The readout counter received from DEC is bigger than the number of readout in a ramp. |
| | |

## 5.5 Packet Content description

### 5.5.1 Nominal Housekeeping

- Nominal housekeeping packets are sent to DPU at regular interval.
- Measures are not compressed (every measure is aligned on a 32bits words).
- Measures are stored in the order defined by the Ids (note that the custom entries are available only in diagnostic mode and are therefore not included in the nominal hk packet).
- Each measure is included in each packet.
- Packets always have the same structure

### 5.5.2 Diagnostic Housekeeping

- Diagnostic housekeeping packets are sent at regular interval (definable by command)
- Their content is definable in the Housekeeping Diagnostic List.
- Measure are "compressed"; they are not aligned on 32bits words.
- The packet structure is defined below :

Consider that the Housekeeping Diagnostic List contains the following Ids :

| Sequence Pointer | Sequencer Options | Sequencer Status | End Of Hk List |
|---|---|---|---|
| 2 bytes | 1 byte | 4 bytes | |

The Housekeeping Diagnostic Packet will be organised as follow :

| Byte | Content |
|---|---|
| 0. | Packet ID = 0x00880000 |
| 1. | |
| 2. | |
| 3. | |
| 4. | Length of Data (in words). This is the total length of the packet without the first 2 words (Packet ID + length). |
| 5. | |
| 6. | |
| 7. | |

| | |
|---|---|
| 8. | OBSID |
| 9. | |
| 10. | |
| 11. | |
| 12. | BBID |
| 13. | |
| 14. | |
| 15. | |
| 16. | Time |
| 17. | |
| 18. | |
| 19. | |
| 20. | |
| 21. | |
| 22. | Number of measures in the list |
| 23. | Number of samples in the packet |
| 24. | DMC_B_SPEC_READ |
| 25. | |
| 26. | |
| 27. | |
| 28. | DMC_R_SPEC_READ |
| 29. | |
| 30. | |
| 31. | |
| 32. | DMC_OBT_COUNT |
| 33. | |
| 34. | |
| 35. | |
| 36. | HK Diag Period:<br>If the value is >0, it is the period in ms between two samples.<br>If the value is <0, it is the detector on which the samples are synchronized (-1=BLUE DEC, -2 = RED DEC, -4 = BOLC) |
| 37. | |
| 38. | |
| 39. | |
| 40. | Housekeeping Diagnostic List ID 0 MSB (=0 in our example) |
| 41. | Housekeeping Diagnostic List ID 0 LSB (=10 in our example) |
| 42. | Housekeeping Diagnostic List ID 1 MSB (=0 in our example) |
| 43. | Housekeeping Diagnostic List ID 1 LSB (=9 in our example) |
| 44. | Housekeeping Diagnostic List ID 2 MSB (=0 in our example) |

| | |
|---|---|
| 45. | Housekeeping Diagnostic List ID 2 LSB (=0 in our example) |
| 46. | Housekeeping Diagnostic List ID 3 MSB (=0xFF in our example) |
| 47. | Housekeeping Diagnostic List ID 3 LSB (=0xFF in our example) |
| 48. | … |
| 49. | |
| 50. | |
| 51. | |
| 52. | |
| 53. | |
| 54. | |
| 55. | |
| 56. | |
| 57. | |
| 58. | |
| 59. | |
| 60. | |
| 61. | |
| 62. | |
| 63. | |
| 64. | |
| 65. | |
| 66. | |
| 67. | |
| 68. | |
| 69. | |
| 70. | |
| 71. | Housekeeping Diagnostic List ID 15 LSB |
| 72. | 1st byte of 1st measure of 1st sample (=MSB of Sequence Pointer in our example) |
| 73. | 2nd byte of 1st measure of 1st sample (=LSB of Sequence Pointer in our example) |
| 74. | 1st byte of 2nd measure of 1st sample (=Sequencer Options in our example) |
| 75. | 1st byte of 3rd measure of 1st sample (=MSB of Sequencer Status in our example) |
| 76. | 2nd byte of 3rd measure of 1st sample (=2nd byte of Sequencer Status in our example) |
| 77. | 3rd byte of 3rd measure of 1st sample (=3rd byte of Sequencer Status in our example) |
| 78. | 4th byte of 3rd measure of 1st sample (=LSB of Sequencer Status in our example) |
| 79. | 1st byte of 1st measure of 2nd sample (=MSB of Sequence Pointer in our example) |
| 80. | 2nd byte of 1st measure of 2nd sample (=LSB of Sequence Pointer in our example) |
| 81. | 1st byte of 2nd measure of 2nd sample (=Sequencer Options in our example) |
| … | … |
| … | … |

# 6 Other information

## 6.1 Time-stamping

This section will contain a description of the various ways to time-stamp various information like science data and housekeeping.

The commands related to time-stamping are:

- DMC_WRT_TIME is used to transfer the time from DPU to DMC. It has no direct effect until the DMC_SET_TIME has been sent.
- DMC_SET_TIME copies the last time that has been written through the DMC_WRT_TIME command and resets:
    - DMC_OBT_COUNT
    - DMC_BOL_READ_CNT
    - DMC_B_SPEC_READ
    - DMC_R_SPEC_READ

### 6.1.1 Photometry science packet

The combination of the Time (TMP) and the OBT counter (CRDC).
OBT counter is a 131072 KHz counter provided by the spacecraft.

### 6.1.2 Spectroscopy science packet

The combination of the Time (TMP) and the number of readouts since last set-time (CRDC)
The readout counter frequency is derived from the OBT by error free divisions. Its frequency is function of the timing parameters that have been sent to the DEC.

### 6.1.3 Nominal housekeeping

The combination of the TIME (DMC_TIME_1 and DMC_TIME_2) and the OBT counter.

### 6.1.4 Diagnostic housekeeping

The combination of the Time (TMP) and the OBT counter (DMC_OBT_COUNT). If the diagnostic housekeeping is synchronised on one of the DEC readout, it is probably more interesting to use the Time and the readout counter for this DEC.
Note : all these values are snapshot taken at the same time as the first sample of the packet.

When synchronized with a detector readout, the hk diag is sampled when the 1355 packet has been received by DMC. Then, all the measures are taken sequentially (in the order defined in the DMC_WRT_DIAG_HK_LIST)

### 6.1.5 Involved commands

#### 6.1.5.1 SetTime

- Sets the time
- Resets the DMC_OBT_COUNT counter (also known as CRDC in photometry packet header).
- Resets the DMC_B_SPEC_READ, DMC_R_SPEC_READ, DMC_BOL_READ_CNT
- Does not change OBSID and BBID

#### 6.1.5.2 Set OBSID

- Modify OBSID only

#### 6.1.5.3 Set BBID

- Modify BBID only

### 6.1.6 Summary of commands and telemetry

**Trigger Commands:**

- 7 DMC_SET_TIME
- 8 DMC_SET_OBSID
- 9 DMC_SET_BBID
- 10 DMC_SYNCHRONIZE_ON_DET
- 11 DMC_SET_TIMING_FPGA_PAR

**Write Commands:**

- 157 DMC_WRT_TIMING_FPGA_PAR

**HK nominal:**

- 224 DMC_OBSID
- 225 DMC_BBID
- 226 DMC_TIME_1
- 227 DMC_TIME_2
- 242 DMC_IRS_CNT
- 453 DMC_OBT_COUNT

**HK diag:**

- none

## 6.2 Synchronization of DMC science header and science data

The array below shows, for each entry of the DMC spectroscopy header, at what time the field is sampled with respect to the science data.

| *Parameter* | *Description / use* | *Sampled* |
|---|---|---|
| OBSID | Observation Identification | When the 1355 packet has been completely received from DEC |
| BBID | Building Block Identification | When the 1355 packet has been completely received from DEC |
| LBL | Label | When the 1355 packet has been completely received from DEC |
| TMP | Timing Parameters | When the 1355 packet has been completely received from DEC |
| VLD | Notifies if the science data is valid (0xff) or invalid (0x00) | When the 1355 packet has been completely received from DEC |
| CPR | Chopper position as encoded by MEC | When the 1355 packet has been completely received from DEC |
| WPR | Wheel position as encoded by MEC | When the 1355 packet has been completely received from DEC |
| GPR | Grating position as encoded by MEC | When the 1355 packet has been completely received from DEC |
| CRCRMP | Current ReaDout Count : Current value of the readouts counter, starts from Nr and decrements, value of 0 signals a destructive readout and the end of an integration interval | This field is extracted from the DEC packet |
| RRR | Readouts in ramp ( Nr ) Readback : Number of readouts within the same integration ramp ( i.e. between successive capacitor resets ) | This field is extracted from the DEC packet |
| CRDC | number of readouts since the last SET_TIME command to DMC | When the 1355 packet has been completely received from DEC |
| CRECR | CRE Control Readback | This field is extracted from the DEC packet |

The array below shows, for each entry of the DMC photometry header, at what time the field is sampled with respect to the science data.

| *Parameter* | *Description / use* | *Sampled* |
|---|---|---|
| OBSID | Observation Identification | When the first 1355 packet has been completely received from DEC |

| | | |
|---|---|---|
| BBID | Building Block Identification | When the first 1355 packet has been completely received from DEC |
| LBL | Label | When the first 1355 packet has been completely received from DEC |
| TMP | Time | When the first 1355 packet has been completely received from DEC |
| VLD | Notifies if the science data is valid (0x000000FF) or invalid (0x00000000) | When the first 1355 packet has been completely received from DEC |
| CPR | Chopper position as encoded by MEC | When the first 1355 packet has been completely received from DEC |
| WPR | Wheel position as encoded by MEC | When the first 1355 packet has been completely received from DEC |
| BOLST | BOLC status word as described in [RD24] | This field is extracted from the first BOLC packet for this readout |
| CRDC | the number of OBT clock ticks since the last SET_TIME command to DMC | This field is sampled when the synchro signal from BOLC is received by the timing FPGA (no software -> no jitter). |
| CRDCCP | Current ReaDout Count in Chopper Position. This counter is reset each time the chopper start moving. | This field is incremented and sampled when the synchro signal from BOLC is received (this is done in the 8KHz interrupt routine). |
| DBID | Data Block ID. Contains the ID of the block of detector arrays whose data are included in this packet.<br>1 = Array 1 and 2<br>2 = Array 3 and 4<br>3 = Array 5 and 6<br>4 = Array 7 and 8<br>5 = Array 9 and 10 | This field is generated by DMC OBSW. For this, DMC assumes the packets are received in the right order. |

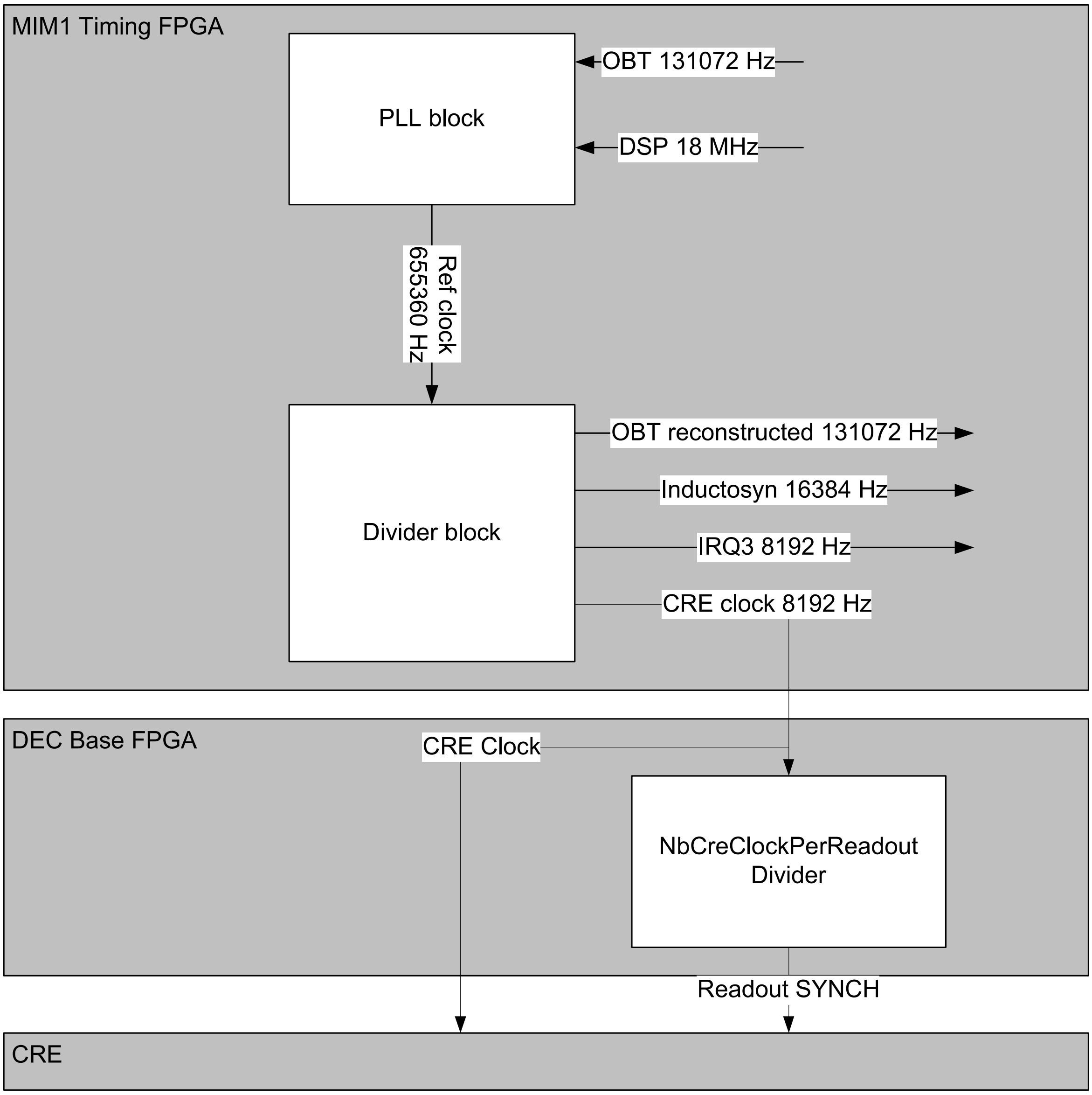
## 6.3 Science data sampling

For deriving the science data sampling, the MIM1 FPGA receives the OBT (131072 Hz) and the DSP clock (18 MHz). From these frequencies, the PLL block produces a reference clock at 655360 Hz. If the OBT is missing, the PLL block will automatically produce the same reference clock from the DSP clock. Note that in this case, if the DSP clock is not exactly 18 MHz, the reference clock will not be exactly 655360 Hz.

Then, this reference clock is divided many times and all internal DMC timings are derived from there. One of this signal, the IRQ3 is triggering the interrupt routine, another one is the CRE clock that is provided to the DEC Base FPGA.

The DEC Base FPGA then provides the CRE clock directly to the CREs and counts the number of CRE clock to produce the readout SYNCH based on the programmed value in NbCreClockPerReadout.

The DEC Base FPGA also counts the NbReadoutPerRamp to produce the Desctructive readout SYNCH.



## 6.4 Detecting Memory Errors

The DSP board contains 2 EDAC chips that can detect memory failure in both DM and PM (including EEPROM). In the case of single bit failures, the EDAC signals the error and correct it. In the case of double bit failures, the EDAC signals the error but it can not be corrected.

The DMC OBS regularly access each memory cell in order to detect errors and to correct the single failures. This piece of code is called 'memory scrubbing'. It is included in the HK nominal task and checks 32 words in DM and 32 words in PM every 2 seconds. It means that DM is completely checked every 9 hours and PM every 14 hours.

4 kinds of errors can be detected:

- Single failure in DM
- Single failure in PM
- Double failure in DM
- Double failure in PM

For each of these errors, the last 256 failing addresses are stored in arrays that are accessible through a dump command. Furthermore, 4 values in the nominal HK identifies where the next failing address will be stored in these arrays.

### 6.4.1 Example

At start-up of the software, all 4 arrays are empty and the 4 indexes are zero. Let's consider the single failure in DM only.

At start-up:

DM_SF_FAILING_ADDRESSES:

| | | | | |
|---|---|---|---|---|
| 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | … |

DMC_DM_SF_IND = 0

After a few seconds of execution, the memory scrubbing function accesses the memory cell 0x00000105 and the EDAC signals an error, the array and index will then be:

DM_SF_FAILING_ADDRESSES:

| | | | | |
|---|---|---|---|---|
| 0x00000105 | 0x00000000 | 0x00000000 | 0x00000000 | … |

DMC_DM_SF_IND = 1

A few seconds later, another task access the memory cell 0x0000F102 and the EDAC signals an error, the array and index will then be:

DM_SF_FAILING_ADDRESSES:

| | | | | |
|---|---|---|---|---|
| 0x00000105 | 0x0000F102 | 0x00000000 | 0x00000000 | … |

DMC_DM_SF_IND = 2

A few hours later, the memory scrubbing function accesses the same memory cell 0x0000F102 and the EDAC signals an error again, the array and index will then be:

DM_SF_FAILING_ADDRESSES:

| | | | | |
|---|---|---|---|---|
| 0x00000105 | 0x0000F102 | 0x0000F102 | 0x00000000 | … |

DMC_DM_SF_IND = 3

…

The array is actually a circular buffer so once it is full, it will overwrite the first elements again and the index will start counting at zero again.

Note: the memory error detection process is a low level process that can not access the DMC_SW_GLOBAL_ST variable. Therefore, it can not signal any error in this variable. The only way to know that a memory error has been detected is to check if the indexes have been modified since the last HK packet.

### 6.4.2 How to react ?

Every time the ground software (or DPU ?) detects that one of the index 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.

Single failures are not critical since the EDAC can correct them. However, if a memory cell has a permanent single failure, it means that double failures are more probable to occur on this cell.

Any permanent failing memory cell (single or double failure) shall be avoided. This is not an easy task to do since it requires the software to be modified such that it does not use the memory cell anymore and such that the memory scrubbing does not test it anymore.

A procedure shall be provided to be able to re-compile a new version of the DMC OBS very fast. Temporary solution shall be to switch to redundant DMC while the new version is being prepared.

### 6.4.3 Related dump commands

Here are the information needed to be able to dump the arrays of failing address. Note that these addresses are subject to change for every new version of the DMC software.

| Name | Memory ID | Start address | length |
|---|---|---|---|
| DM_SF_FAILING_ADDRESSES | DRAM | | 256 words |

| | | | |
|---|---|---|---|
| PM_SF_FAILING_ADDRESSES | DRAM | | 256 words |
| DM_DF_FAILING_ADDRESSES | DRAM | | 256 words |
| PM_DF_FAILING_ADDRESSES | DRAM | | 256 words |

### 6.4.4 Summary of commands and telemetry

**Trigger Commands:**

- none

**Write Commands:**

- none

**HK nominal:**

- 455 DMC_DM_SF_IND
- 456 DMC_PM_SF_IND
- 457 DMC_DM_DF_IND
- 458 DMC_PM_DF_IND

**HK diag:**

- none