Test Report

DMC OBS v6.028 Functional Test Report

PACS-CL-TR-044, issue 1.3
10 November 2008

Prepared by	A. Mazy
Verified by	E. Renotte
Released by	E. Renotte
Approved by	H. Feuchtgruber

DRD : ECSS-E-10-02A – Annex G

Document Change Record

Issue	Date	Changes
Issue 1.0	05 April 2007	initial release. This issue includes the results from the test run on: • The 14th of February 2007-04-05 • November 2006 (Calibration source long duration tests) • August 2006 (CAL Data calibration)
Issue 1.1	14 june 2008	Test run on the DMC OBS v6.023
Issue 1.2	09 oct 2008	Test run on the DMC OBS v6.026 These scripts have been adapted: A_bb2.txt, a_synchro.txt, a_write1.txt, a_bol2spu2.txt, a_bol2spu.txt, a_dec2spu2.txt, a_dec2spu.txt, a_fw.txt, a_mec.txt
Issue 1.3	10 nov 2008	Test run on the DMC OBS v6.028 Added a_mec_sync.txt to test the shifted sync to trigger the mechanisms move

Distribution List

(Not restricted)

Table of Contents

List of Figures

1 Scope and applicability

1.1 Scope

This test procedure defines the instructions for conducting the DEC/MEC PFM functional and performance tests of the Herschel PACS project.

This test procedure is based on the requirements of the DEC/MEC AIV Plan [AD06], as amended per [RD06].

In the scope of the delivery of the DMC OBS v6.026, only a subset of the PFM functional test has been run.

1.2 Purpose

This test procedure defines in detail the test and the corresponding “as-run” procedure will become part of the test report.

2 References

2.1 Applicable documents

The following documents are applicable and are referred to as [AD**] in the text :

[AD01]	ESA PT-IID-A-04624	Herschel/Planck Instrument Interface Document - Part A (issue 3.3)
[AD02]	ESA PT-RQ-04410	PA Requirements for FIRST/PLANCK Scientific Instruments (issue 2)
[AD03]	PACS-ME-PL-007	PACS Project Product Assurance Plan (issue 2)
[AD04]	PACS-CL-RS-003	DEC/MEC Requirement Specification
[AD05]	PACS-CL-RS-001	DEC/MEC User Requirement Document – URD
[AD06]	PACS-CL-PL-003	DEC/MEC AIV Plan

2.2 Reference documents

The following documents are referenced for supporting information and are referred to as [RD**] in the text :

[RD01]	ECSS-E-10-02A	Verification
[RD02]	ECSS-E-10-03A	Testing
[RD03]	PACS-CL-SR-002	DEC/MEC Software User Manual - SUM
[RD04]	PACS-CL-TP-021	DEC/MEC PFM Inspection Procedure
[RD05]	OIP/767-052_B	DEC/MEC MICD
[RD06]	PACS-CL-PL-012	PACS DEC/MEC PFM Engineering Test Plan
[RD07]	PACS-CL-TP-016	PACS DEC/MEC QM Acceptance Test Plan
[RD08]	PACS-CL-TR-028	PACS DEC/MEC EQM Functional Test Report
[RD09]	TBD (OIP)	DEC/MEC Packing, Handling & Installation Procedure

3 Definitions and abbreviations

3.1 Definitions

Acceptance A verification stage with the objective of demonstrating that the product is free of workmanship defects and integration errors and ready for its intended use.

Inspection A verification method that determines conformance to requirements for constructional features, document and drawing conformance, workmanship and physical conditions without the use of special laboratory equipment, procedures or services.

Qualification The verification stage with the objective to demonstrate that the design meets the applicable requirements including proper margins.

Test A verification method wherein requirements are verified by measurement of product performance and functions under various simulated environments.

3.2 Abbreviations

ADC	Analog-to-Digital Converter
AIV	Assembly Integration & Verification
AR	Acceptance Review
AVM	Avionic Verification Model
ASW	Application SoftWare
BOL	Beginning of Life
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
DMC	DEC/MEC
DPU	Digital Processing Unit
ECR	Engineering Change Request
EEPROM	Electrically Erasable PROM
EIDP	End Item Data Package
EGSE	Electrical Ground Support Equipment
EM	Engineering/Electrical Model
EMC	Electromagnetic Compatibility
EOL	End of Life
EQM	Electrical/Engineering Qualification Model
ESD	Electrostatic Discharge
FEE	Front End Electronics
FM	Flight Model
FPGA	Field Programmable Gate Array
FPU	Focal Plane Unit
FS	Flight Spare
GSE	Ground Support Equipment
HK	HouseKeeping
H/W	Hardware
ICD	Interface Control Document
I/F	Interface
IID-A	Instrument Interface Document - Part A
IID-B	Instrument Interface Document - Part B
LISN	Line Impedance Simulation Network
NA	Not Applicable
NCR	Nonconformance Report
NRB	Nonconformance Review Board
OBS	On-Board Software
PACS	Photodetector Array Camera and Spectrometer

PFM	Proto- Flight Model
PI	Prime Investigator
PROM	Programmable ROM
PTR	Post-Test Review
QM	Qualification Model
RAM	Random Access Memory
ROM	Read-Only Memory
S/C	SpaceCraft
SFT	Short Functional Test
SPU	Signal Processing Unit
S/S	Sub-System
SSD	Software Specification Document
STM	Structural-Thermal Model
SUM	Software User Manual
SUSW	StartUp SoftWare
S/W	Software
TBC	To Be Confirmed
TBD	To Be Defined
TRB	Test Review Board
TRR	Test Readiness Review
TBW	To Be Written
URD	User Requirement Document

4 Requirements to be verified – Test requirements

This section lists the requirements to be verified in the specific test and provides traceability where in the test the requirement is covered.

- Processor budget (§.9.3.10)
- Start-up, initialisation, commandability and shut down (§.9.3.2.1 & §.9.3.2.2)
- Availability of nominal and diagnostic housekeeping (§.9.3.3)
- Commands verification (§.9.3.4)
- Commands sequence handling (§.9.3.5.1)
- Availability and selectability of mechanisms synchronisation sources (§.9.3.5.2)
- Availability and performance of the Grating mechanism control functions (§.9.3.5.3, 9.3.13.1 and 9.3.13.3)
- Availability and performance of the Chopper mechanism control functions (§.9.3.5.4 and 9.3.13.1)
- Availability and performance of the Filter Wheel mechanisms control functions (§.9.3.5.6)
- Availability and performance of the Calibration Sources control functions (§.9.3.5.5)
- Availability and performance of the Temperature Sensors acquisition (§.9.3.6)
- Availability of the SPU analogue housekeeping (§.9.3.15)
- Availability and performance of the detector Heater control functions (§.9.3.7.1.1)
- Availability and performance of the detector Flasher control functions (§.0)
- Availability and performance of photoconductor arrays control, data acquisition and data handling functions (§.0, 9.3.13.2, and 9.3.14)
- Availability of bolometer arrays control, data acquisition and data handling functions (§.9.3.7.3)
- Timing and synchronisation performances (§.9.3.8)

5 Test article

5.1 Identification and Configuration of the Test Article

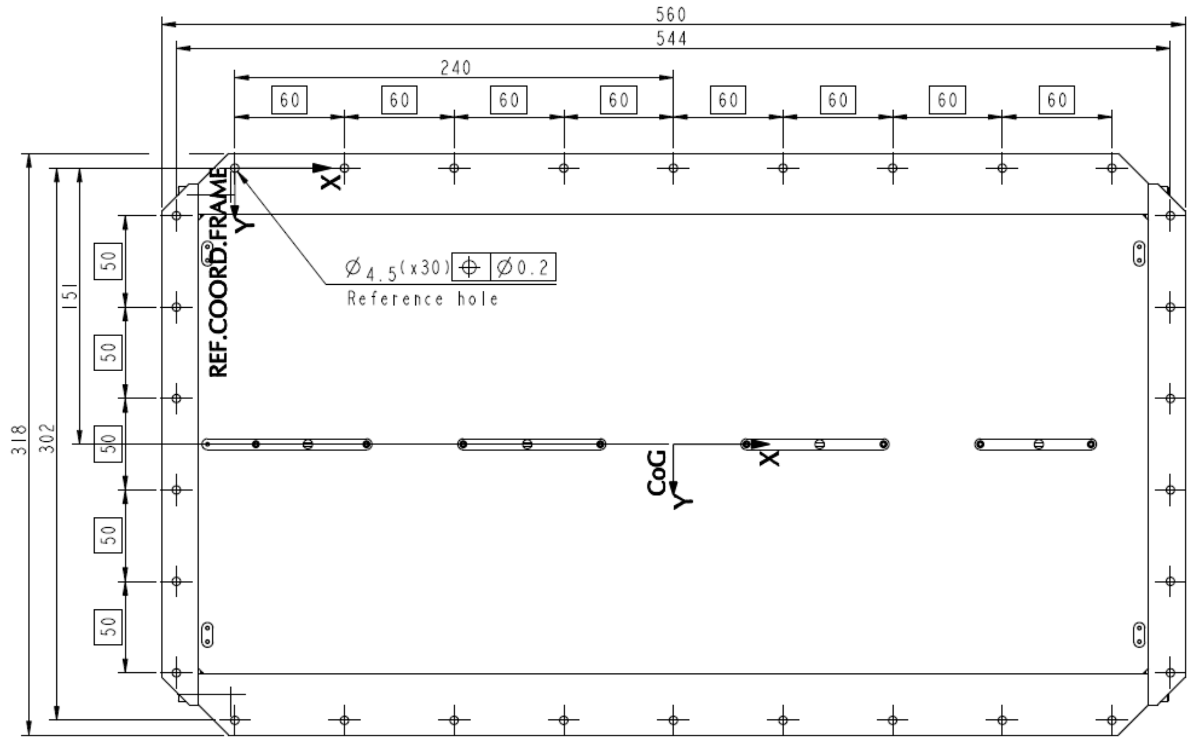
This procedure is for the following test article:

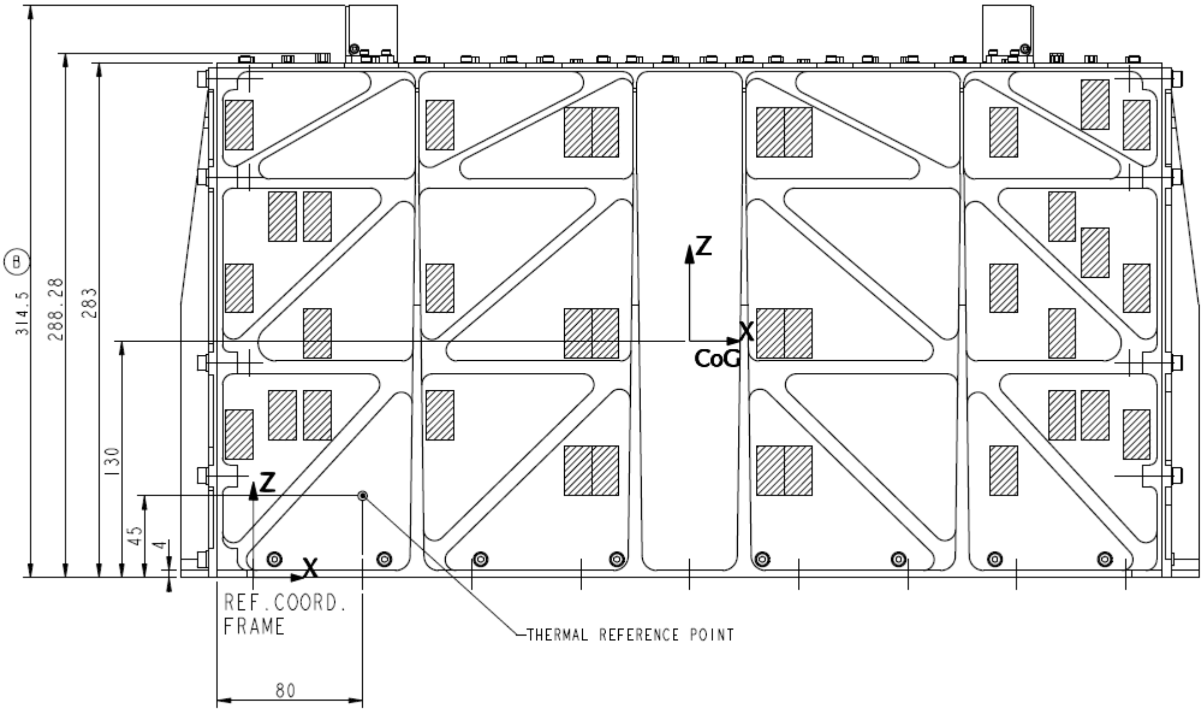
Programme:	Herschel - PACS
Contract Nr:	N/A
WBS / Product Tree Nr:	N/A
Unit:	Detectors & Mechanisms Controller (DEC/MEC)
Model:	Flight Model, software v6.028
Serial Number:	N/A
Supplier:	Centre Spatial de Liège (Liege, Belgium)
Date of construction:	2006 (software: 9th oct 2008)

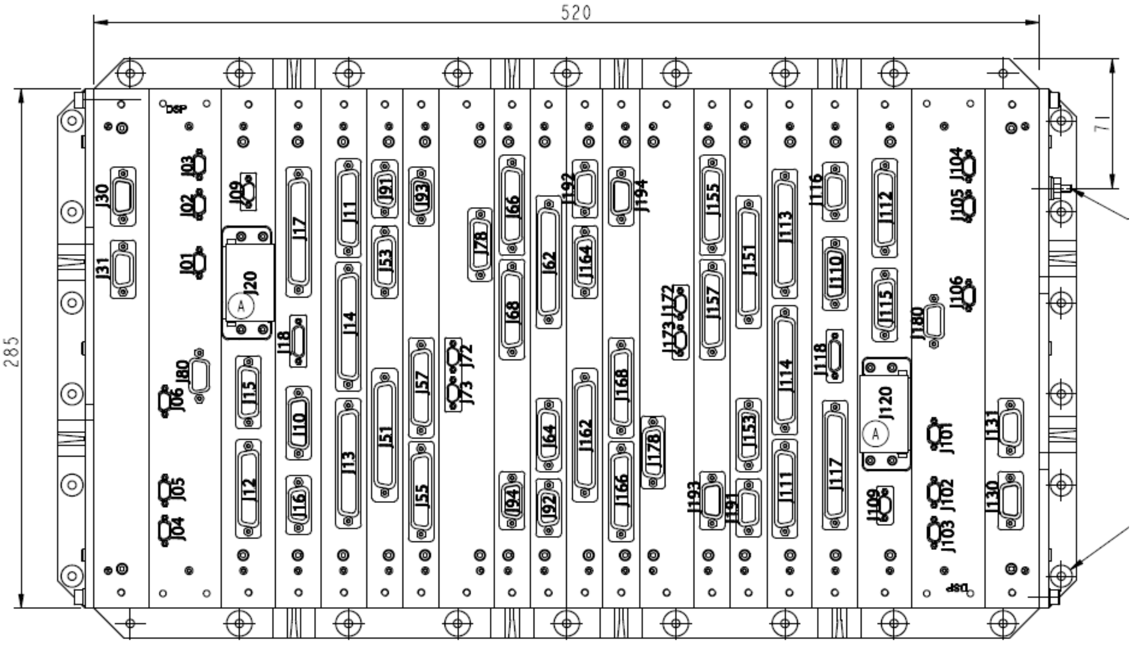
For the execution of this test, the test article shall be in flight-like configuration.

5.2 Equipment dimensions & reference axes

Figure 5-1: DEC/MEC general dimensions & reference axes.




314.5
288.28
283
130
45
4
B
Z
X
CoG
Z
X
REF.COORD.
FRAME
THERMAL REFERENCE POINT
80


520
285
71
DSP
J30
J31
J03
J02
J01
J09
J20
A
J80
J06
J05
J04
J17
J18
J15
J12
J10
J16
J11
J14
J13
J91
J53
J51
J93
J57
J55
J78
J73 J72
J66
J68
J94
J62
J64
J92
J192
J164
J162
J194
J168
J166
J178
J173 J172
J193
J155
J157
J151
J153
J191
J113
J114
J111
J116
J110
J118
J117
J112
J115
J120
A
J109
J105 J104
J106
J180
J101
J102
J103
J131
J130

5.3 Operating Modes

All operating modes of the DEC/MEC will be used during the test.

6 Test set-up

6.1 Test configuration

Figure 6-1: Functional test set-up (nominal configuration).

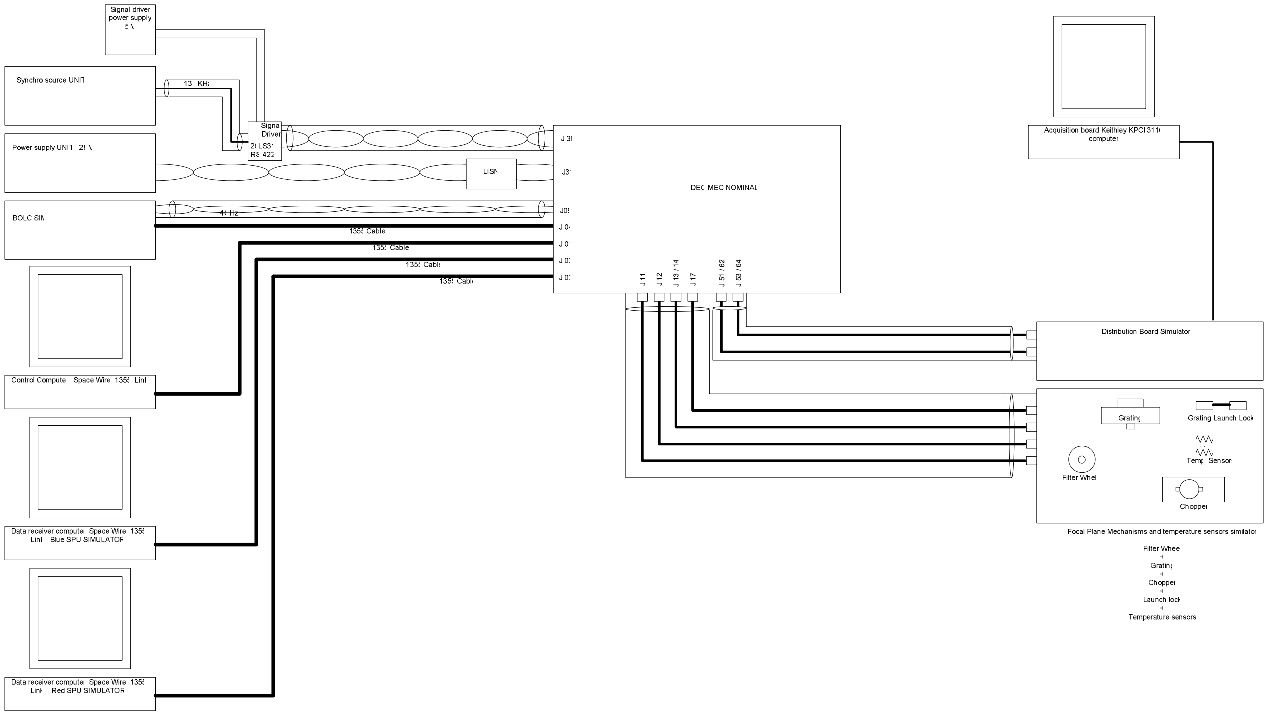
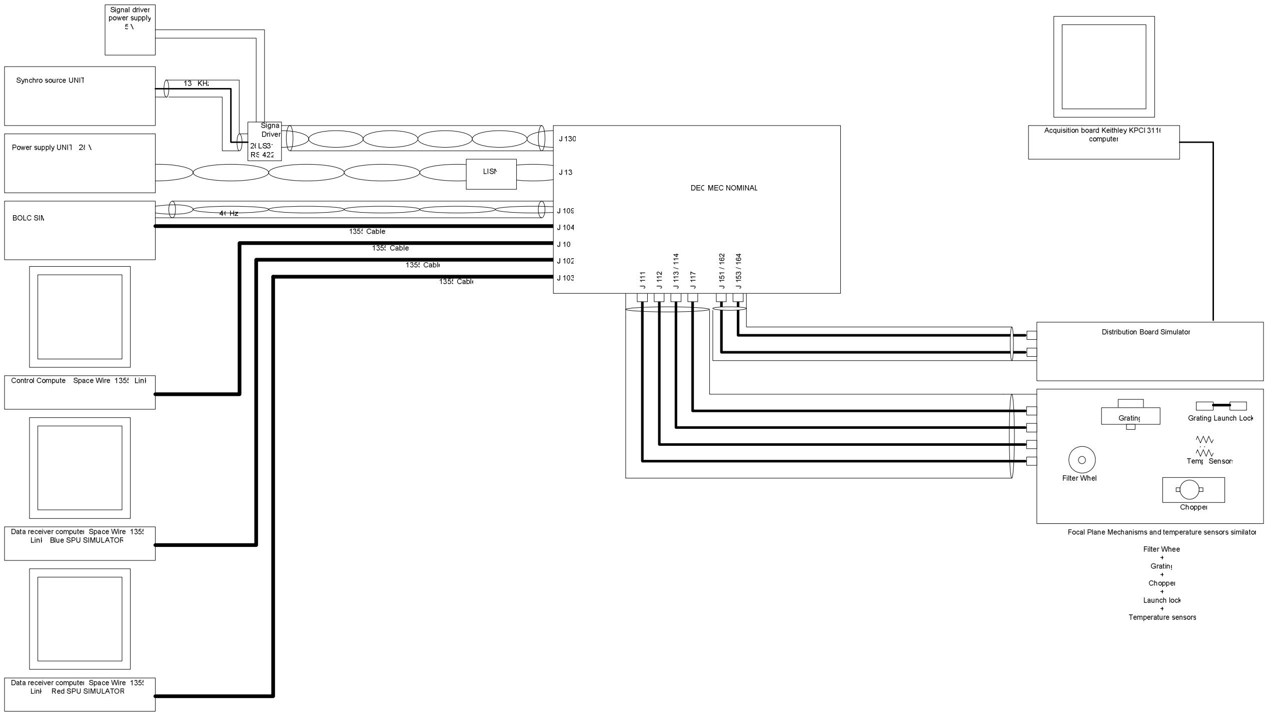

Signal driver power supply 5 V
Synchro source UNIT
13 KHz
Signal Driver 26LS31 RS 422
Power supply UNIT 28 V
LISN
BOLC SIM
40 Hz
1355 Cable
1355 Cable
1355 Cable
1355 Cable
J 30
J31
J09
J 04
J 01
J 02
J 03
DEC MEC NOMINAL
J 11
J 12
J 13 / 14
J 17
J 51 / 62
J 53 / 64
Control Computer Space Wire 1355 Link
Data receiver computer Space Wire 1355 Link Blue SPU SIMULATOR
Data receiver computer Space Wire 1355 Link Red SPU SIMULATOR
Acquisition board Keithley KPCI 3116 computer
Distribution Board Simulator
Grating
Grating Launch Lock
Temp Sensors
Filter Whel
Chopper
Focal Plane Mechanisms and temperature sensors similator
Filter Wheel
+
Grating
+
Chopper
+
Launch lock
+
Temperature sensors

Figure 6-2: Functional test set-up (redundant configuration).


Signal driver power supply 5 V
Synchro source UNIT
13 KHz
Signal Driver 26LS31 RS 422
Power supply UNIT 28 V
LISN
BOLC SIM
40 Hz
1355 Cable
1355 Cable
1355 Cable
1355 Cable
J 130
J 13
J 109
J 104
J 10
J 102
J 103
DEC MEC NOMINAL
J 111
J 112
J 113 / 114
J 117
J 151 / 162
J 153 / 164
Control Computer Space Wire 1355 Link
Data receiver computer Space Wire 1355 Link Blue SPU SIMULATOR
Data receiver computer Space Wire 1355 Link Red SPU SIMULATOR
Acquisition board Keithley KPCI 3116 computer
Distribution Board Simulator
Grating
Grating Launch Lock
Temp Sensors
Filter Whel
Chopper
Focal Plane Mechanisms and temperature sensors similator
Filter Wheel
+
Grating
+
Chopper
+
Launch lock
+
Temperature sensors

6.2 Test site conditions and monitoring

The test shall be performed in ambient/vacuum conditions:

- Ambient temperature: 20°C ± 2°C
- Relative humidity: 55% ± 10%
- Cleanliness: Class 100,000 or better
- ESD: TBD

Ambient conditions shall be monitored.

6.3 Data handling

TBW

6.4 Ground Support Equipment (GSE)

The following GSE are developed for this test (see Annex 1):

- EGSE PC equipped with a SpaceWire board and the following test softwares:
  - SimDPUv2.6 modified by CSL to have a better interface to send commands to DMC.
  - A basic Link Receiver that stores all the packets received on disk.
  - Sequence Writer is a program to create sequences.
  - DiagHkRTViewer is a graphical display for the diagnostic housekeeping.
  - “view hk.bat” is a script to compile the nominal housekeeping packets into an Excel file.
  - “view hk as text.bat” is a script to compile the nominal housekeeping packets into a text file.
  - “view hkDiag.bat” is a script to compile the diagnostic housekeeping packets into an Excel file.
- SPU Simulators (blue and red)
- Controlled switch (MOSFET Switch)
- LISN
- FPU mechanism simulators (Grating, Chopper, Filter wheel, Grating launch-lock)
- BOLC Simulator
- Distribution board simulator
- Temperature sensor simulators
- Calibration source simulator

All test software are under source control management in a Visual Source Safe database. The versions used during FM campaign are labelled ‘FM acceptance”.

6.5 Test equipment & instrumentation

The following equipment will be used during the test (see Annex 2):

Equipment	Description / Specification	Calibrated until
Laboratory power supply	SkyTronic 650-682 0-30V 0-10A	NA
Current probe	Tektronix A 6302 DC-50 MHz	NA
Current probe amplifier	Tektronix AM 503	NA
Digital oscilloscope	Lecroy WAVEPRO 940	November 2006
Digital multi-meter	Datron	04/08/2007
Acquisition board	Keithley KPCI-3116	Auto-calibrated

6.6 Test harness

See Figure 6-1 and Figure 6-2.

7 Personnel required and responsibilities

This procedure shall be executed by skilled operators only !

Function	Name	Ext.	Mobile
Test Responsible:	Eric CALLUT	340	
Project Manager:	Etienne RENOTTE	300	+32 477 50 28 23
PA/QA Manager:	Michel THOME	325	
System Engineer:	Jean-Marie GILLIS	338	
Software Engineer:	Alain MAZY	342	
Electrical Engineer:	Francis MONTFORT	408	
Agency Representative:	TBD	-	
Customer Representative:	Bernhard VOSS (MPE)	-	

8 Test constraints and operations

8.1 Operational constraints

As per applicable QA standards.

8.2 Special conditions and hazards

As per applicable QA standards.

8.3 Procedure change management

As per applicable QA standards.

8.4 Reporting

Before the test campaign, a "Test Readiness Review" (TRR) shall be organised. The purpose of the TRR is to verify that the test article, the test facility, the test equipment and the relevant support documentation are ready to start the test. Furthermore all open non-conformances shall be reviewed to check whether they may affect the test. A check-list for the TRR is given in §.10.3.

After the test campaign, a "Post-Test Review" (PTR) shall be organised. The purpose of the PTR is to verify that all the test objectives were met or attempted to maximum extent. A check-list for the PTR is given in §.10.4.

8.5 Anomaly management

Failures, non-conformances or other anomalies observed during the test on the test article, test equipment or test facility shall be factually recorded in the appropriate log book as soon as they are detected and immediately reported to the Test Responsible. All anomalies shall be investigated to the maximum extent before proceeding to the next step of the test procedure. As required, the Test Responsible and the PA/QA Manager will issue an NCR to be processed according to the applicable PA/QA standards. When needed, the Project Manager shall take the responsibility for reporting non-conformances to the Customer and/or Agency representative.

8.6 PA/QA aspects

See [AD03].

9 Step-by-step instructions

9.1 Test sequence overview

The functional and performance test of the DEC/MEC will be conducted according to the following sequence:

- Setting up
- Power characteristics
- Switch On/Off
- Housekeeping
- DPU interface
- Instrument control
- Timing
- Temperature sensors
- Detector control
- Timing
- Interface
- Resource
- Reliability
- Trigger commands
- Write commands
- CRE interface
- SPU analogue interface
- Short functional test
- Dismounting

The procedure PACS-CL-TP-022 has been used. This document is actually a filled version of this procedure.

9.2 Test preparation

This test is not applicable to a 'software only' acceptance test

Step	Activity Description	Expected Outcome	Actual Result	Conductor	Control	Remarks
2001	Verify the conformance of the test area environmental parameters, as defined in §.6.2, and the readiness of environmental monitors					
2002	Check electrical continuity of groundings, ESD cloths and wristlets, turn on active ESD protections (as required)					
2003	Check the readiness of GSE, instrumentation and harnesses defined in §.6.4, 6.5 and 6.6					
2004	Unpack DEC/MEC according to OIP procedure [RD09]					
2005	Install DEC/MEC in test area on a clean and flat ESD cloth					
2006	Dismount handles according to OIP procedure [RD09]					
2007	(As necessary – always needed after transport) conduct inspection of the DEC/MEC according to [RD04]					
2008	Connect the grounding strap, lock at nominal torque (2.2 Nm ± 10%)					
2009	Install the local harness as shown in **Error! Reference source not found.** (or verify the connections if already installed) ; check all connectors (cleanliness, pin alignment) before mating ; lock fasteners at nominal torques (micro-miniature connectors: TBD Nm, sub-miniature connectors: TBD Nm)					
2010	Connect the DEC/MEC to GSE and test equipment as shown in Figure 6-1 (or verify the connections if already installed) ; check all connectors (cleanliness, pin alignment) before mating ; lock fasteners at nominal torques (micro-miniature connectors: TBD Nm, sub-miniature connectors: TBD Nm)					

9.3 Test performance (step-by-step procedure)

The following procedure is used routinely to start up the DEC/MEC, establish the communication with the EGSE and to start a new test:

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3001	Start up the EGSE (PC)					
3002	Switch on the laboratory power supply, adjust to 28V					
3003	Switch on the DEC/MEC with the controlled switch					
3004	Wait 10 seconds					
3005	From the EGSE, start “Start from EEPROM.bat”					
3006	Wait a few seconds for the SIM DPU menu display					

Most of the time, this test procedure will be supported by **test scripts** that are executed by the DPU simulator on the EGSE. Test scripts and their language are presented in Annex 3.

9.3.1 Power characteristics

This test is not applicable to a 'software only' acceptance test

9.3.2 Switch On/Off

9.3.2.1 Switch On (SWON.1)

This test is to verify that the DEC/MEC is starting up properly.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3201	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		ok			
3202	Switch on DEC/MEC and EGSE (see above)		ok			
3203	Check that the mechanisms are not moving during power on	*No move*	ok			
3204	Execute test script **a_swon.txt**, perform actions as prompted by the test script, look at mechanism simulators		*ok*			

3205	Check that DEC/MEC is connected to DPU Simulator	*ChkBit(198, 19, 0)*
		ChkBit(199, 19, 0)
			ok			

3206	Check that blue DEC, red DEC, Chopper and Grating are powered on	*ChkBit(201, 20, 1)*
		ChkBit(204, 20, 1))
		ChkBit(209, 20, 1)
		ChkBit(208, 20, 1)
			ok			

3207	Switch off DEC/MEC		ok			
			7/7			

9.3.2.2 Switch Off (SWOF.1)

This test is to verify that the DEC/MEC is shutting down properly.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3208	Switch on DEC/MEC and EGSE (see above)		ok			
3209	Execute test script **a_swof.txt**, perform actions as prompted by the test script, look at mechanism simulators		ok			
3210	Check that all subsystems are in default state		ok			

3211	Check that blue DEC, red DEC, Chopper and Grating are powered off	*ChkBit(201, 20, 0)*
		ChkBit(204, 20, 0)
		ChkBit(209, 20, 0)
		ChkBit(208, 20, 0)
			ok			

			4/4			

9.3.3 Housekeeping

9.3.3.1 Nominal housekeeping (HKN.1)

This test is to verify that the DEC/MEC is delivering properly the nominal housekeeping data to the DPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3301	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		ok			
3302	Switch on DEC/MEC and EGSE (see above)		ok			
3303	Execute test script **a_hk.txt**, perform actions as prompted by the test script		ok			
3304	Check in directory 'd:\prj\pacs\simulators\simdpuv26\asw' that a file 'hk.dat' has been created and is updating regularly		ok			
3305	Check that 'Hk.txt' contains the whole set of housekeeping variables		ok			
			6/6			

9.3.3.2 Diagnostic housekeeping (HKD.1)

This test is to verify that the DEC/MEC is capable of performing diagnostic monitoring at 200Hz and delivering properly the diagnostic housekeeping data to the DPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3306	Execute test script **a_hkDiag.txt**, perform actions as prompted by the test script		ok			
3307	Check in directory 'd:\prj\pacs\simulators\simdpuv26\asw' that file 'hkDiag.dat' has been updated		ok			
3308	Check that HkDiag.xls contains the 2 variables sampled at 200Hz		ok			
3309	Save file 'a_hkDiag_nom.xls', (resp. 'a_hkDiag_rep.xls')					
			3/3			

9.3.4 DPU interface (DPU.1)

This test is to verify that the DEC/MEC is accepting valid commands from DPU, respectively rejecting invalid commands from DPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3401	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Ok			
3402	Switch on DEC/MEC and EGSE (see above)					
3403	Execute test script **a_dpu.txt**, perform actions as prompted by the test script		Ok			
3404	Check that DPU as received a "PACK" after a valid command		Ok			
3405	Check that DPU as received a "NACK" after an invalid command		Ok			
3406	Switch off DEC/MEC		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
			2/2			

9.3.5 Instrument control

9.3.5.1 Sequences (IC.1)

This test is to verify that the DEC/MEC is able to store operation sequences and execute them later on upon trigger. This test is also to demonstrate that the only command that is accepted during a sequence execution is the DMC_ABORT_SEQUENCE.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3501	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Ok			
3502	Switch on DEC/MEC and EGSE (see above)		Ok			
3503	Execute test script **a_seq.txt**, perform actions as prompted by the test script		Ok			
3504	When synchronised on blue DEC, check in 'Hk.xls' that: - DMC_SEQ_LOOP_ID0 is decreasing from 5 to 1 - DMC_SEQ_LOOP_ID1 is decreasing from 2 to 1, 5 times - DMC_SEQ_WAIT_IND is increasing from 0 to 160, 10 times		Ok			
3505	When synchronised on BOLC, check in 'Hk.xls' that: - DMC_SEQ_LOOP_ID0 is decreasing from 5 to 1 - DMC_SEQ_LOOP_ID1 is decreasing from 2 to 1, 5 times - DMC_SEQ_WAIT_IND is increasing from 0 to 160, 10 times Note that when synchronized on BOLC,		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	DMC_WAIT_IND increases faster					
3506	Check that DPU received a “PACK” after command ‘DMC_ABORT_SEQUENCE’		OK			
3507	Check that DPU received a “NACK” after another trigger command (i.e. change synchronisation source)		OK			
3508	Check that DPU received a “NACK” after a write command (i.e. change SPU transmission mode)		OK			
3509	Check in ‘Hk.xls’ that DMC_B_SPU_TR_MODE has not been modified after the write command attempt		OK			
3510	Save file ‘a_seq_nom.xls’ (resp. ‘a_seq_red.xls’)		OK			
3511	Switch off DEC/MEC		OK			
			Score : 6/6			

9.3.5.2 Mechanisms control (ICM.1)

This test is to verify that:

- mechanism control interrupt service routine is running
- mechanism movements are synchronized with DEC

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3512	Switch on DEC/MEC and EGSE (see above)		Yes			
3513	Execute test script **a_mec.txt**, perform actions as prompted by the test script		Yes			
3514	In ‘Hk.xls’, check that DMC_IRS_CNT is incrementing by 16384, which shows that the mechanism control interrupt routine is running		OK			
3515	Check that the Chopper is moving directly when it is configured to move without synchro		OK			
3516	Check that the Chopper is not moving when the selected synchronisation source (blue DEC) is not active		OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3517	Check that the Chopper starts moving when the selected synchronisation source (blue DEC) is turned on		OK			
3518	Save file ‘a_mec_nom.xls’ (resp. ‘a_mec_red.xls’)		Yes			
			Score : 9/9			

9.3.5.3 Grating mechanism control

9.3.5.3.1 Launch-lock (GRAT.1)

This test is to verify the availability of the Grating Launch-lock (LL) functions.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3519	Connect the LL simulator (both redundant and nominal sides) Execute test script **a_grat_ll.txt**, perform actions as prompted by the test script. The Launch-lock position sensor shall be activated by hand.		Yes			
3520	When requested, on the LL simulator, put the ‘close switch’ ON [down] and the ‘open switch’ OFF [up]					

3521	Check visually that LL is not moving and both switches are at 0 when LL is not moving	*ChkBit(208, 19, 0)*
		ChkBit(208, 30, 0)
		ChkBit(208, 31, 0)
			Ok			

3522	After unlock command, check visually that LL is moving and switches are reporting it is 'locked'	*ChkBit(208, 19, 1)*
		ChkBit(208, 30, 1)
		ChkBit(208, 31, 0)
			Ok			

3523	When requested, on the LL simulator, put the ‘close switch’ OFF [up] and the ‘open switch’ ON [down]		Ok			

3524	After lock command, check visually that LL is moving and switches ‘unlocked’	*ChkBit(208, 19, 1)*
		ChkBit(208, 30, 0)
			Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
		ChkBit(208, 31, 1)				
3525	When requested, on the LL simulator, put the 'close switch' OFF [up] and the 'open switch' OFF [up]		Ok			

3526	After lock command, check visually that LL is moving and switches neither locked nor unlocked	*ChkBit(208, 19, 1)*
		ChkBit(208, 30, 0)
		ChkBit(208, 31, 0)
			Ok			

3527	When requested, on the LL simulator, disconnect actuator 2 (connected to redundant MIM)		NO			No redundant MIM on QM
3528	Check that LL is moving twice in alternate directions		Ok			
3529	Check that LL has not moved		Ok			
3530	When requested, on LL simulator, reconnect actuator 2		No			No redundant MIM on QM
3531	Check that LL is moving twice in alternate directions. The LL should have moved faster.		NO			No redundant MIM on QM
3532	Save 'a_grat_ll_nom.xls' (resp. 'a_grat_ll_red.xls')		Ok			
			Score : 17/18			

9.3.5.3.2 Closed-loop control (GRAT.2)

This test is to verify the availability of the Grating mechanism control functions, including homing.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3533	Execute test script **a_grat.txt**, perform actions as prompted by the test script		Yes			

3534	After enabling the Grating controller, check that the homing has not been performed and is not running presently	*ChkBit(208, 28, 0)*
		ChkBit(208, 29, 0)
			OK			

3535	Try homing the grating with invalid parameters	*DPU receives a NACK (2 times)*	OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3536	After homing command [Trig(44, 1, 0)], check the homing is in progress	*ChkBit(208, 28, 1)*	OK			

3537	Check that the homing has completed	*ChkBit(208, 28, 0)*
		ChkBit(208, 29, 1)
			OK			

3538	After absolute-move command [Trig(42, 1, 0x40000)], check that the Grating has reached the target position	*ChkGT(250, 0x3FFF0)*
		ChkLT(250, 0x40010)
			OK			

3539	After relative-move command [Trig(43, 1, 0x10000)], check that the Grating has reached the target position	*ChkGT(250, 0x4FFF0)*
		ChkLT(250, 0x50010)
			NO			Grating is slightly out of the range because the phase adjustment of the inductosyn were not optimum for STM grating

3540	Check that the Grating has been switched off and that the position readout is invalid	*ChkBit(208, 20, 0)*
		ChkEQ(250, -1)
			OK			

3541	(Grating is switched on but its controller is disabled.) Check that the limit switch is not activated	*ChkBit(208, 23, 0)*	OK			
3542	Press manually the Grating against the nominal limit switch and check that the limit switch is activated	*ChkBit(208, 23, 1)*	OK			
3543	Release then click OK and check the limit switch is no longer activated	*ChkBit(208, 23, 0)*	OK			
3544	Save ‘a_grat_nom.xls’ (resp. ‘a_grat_red.xls’)		Yes			
			Score : 15/16			

9.3.5.3.3 Open-loop control (GRAT.3)

This test is to verify the availability of the Grating degraded (open-loop) control mode.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3545	Execute test script **a_grat_open.txt**, perform actions as prompted by the test script		Yes			

3546	(Grating is switched on in closed-loop control mode)
	Check in DMC_GRAT_CTRL_ST that the Grating is in closed loop and not in degraded mode
		ChkBit(208, 21, 1)
		ChkBit(208, 24, 0)
			OK			

3547	Verify it is not possible to enter into open-loop control mode [Trig(45, 1, 0)]	*DPU received a NACK*	OK			

3548	Check in DMC_GRAT_CTRL_ST that the Grating is in closed loop and not in degraded mode	*ChkBit(208, 21, 1)*
		ChkBit(208, 24, 0)
			OK			

3549	After switch off, check in DMC_GRAT_CTRL_ST that the Grating is not in closed loop and not in degraded mode	*ChkBit(208, 21, 0)*
		ChkBit(208, 24, 0)
			OK			

3550	After switch on and open-loop command, check in DMC_GRAT_CTRL_ST that the Grating is not in closed loop and is in degraded mode	*ChkBit(208, 21, 0)*
		ChkBit(208, 24, 1)
			OK			

3551	Verify it is not possible to enter into closed-loop control mode [Trig(40, 0, 0)]	*DPU received a NACK*	OK			
3552	(The Grating is moved absolute [Trig(42, 1, 0x1000)], then twice relative [Trig(43, 1, 0x1000); Trig(43, 1, -0x1000)]) Check visually that the position after the third move is close to the position after the first move		OK			

3553	After open-loop control switch off, check in DMC_GRAT_CTRL_ST that the Grating is not in closed loop and not in degraded mode	*ChkBit(208, 21, 0)*
		ChkBit(208, 24, 0)
			OK			

3554	Save file 'a_grat_open_nom.xls' (resp. 'a_grat_open_red.xls')		OK			
			15/15			

9.3.5.3.4 Grating Short Functional Test (GRAT.4)

This test shows a nominal operation of the grating

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3555	Execute test script **func_grat.txt**, perform actions as prompted by the test script		Ok			
3556	(This sequence is automatic. It includes homing, sequence of moves and abort.)		Ok			

9.3.5.4 Chopper mechanism control

9.3.5.4.1 Chopper controller (CHOP.1)

This test is to verify the availability of the Chopper mechanism control functions.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3557	Execute test script **a_chop.txt**, perform actions as prompted by the test script		Yes			

3558	(a) Spectroscopy (Chopper synchronised on blue DEC)
	After enabling the Chopper controller, check status bits in DMC_CHOP_CTRL_ST
		ChkBit(209, 20, 1)
		ChkBit(209, 21, 1)
			OK			

3559	Verify that Chopper position is close to zero	*ChkGT(244, -300)*
		ChkLT(244, 300)
			OK			

3560	Chopper is commanded to position 4000. Check that Chopper position is between 3950 and 4050	*ChkGT(244, 3950)* *ChkLT(244, 4050*	OK			

3561	Chopper is moved by -4000 (relative). Check that Chopper position is between -50 and 50	*ChkGT(244, -50)*
		ChkLT(244, 50)
			OK			

3562	Chopper is commanded to 4000 with dither. Check that Chopper position is close to 4000	*Chopper pos is close to 4000 (there has been some dither added)*	OK			
3563	Chopper is moved by -4000 with dither. Check that Chopper position is close to 0.	*Chopper pos is close to 0 (there has been some dither added)*	OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks

3564	After the switch off, check status bits in DMC_CHOP_CTRL_ST	*ChkBit(209, 20, 0)*
		ChkBit(209, 21, 0)
			OK			

3565	(b) Photometry (Chopper synchronised on BOLC)
	After enabling the Chopper controller, check status bits in DMC_CHOP_CTRL_ST
		ChkBit(209, 20, 1)
		ChkBit(209, 21, 1)
			OK			

3566	Verify that Chopper position is close to zero	*ChkGT(244, -300)*
		ChkLT(244, 300)
			OK			

3567	Chopper is commanded to position -4000. Check that Chopper position is between -3950 and -4050	*ChkLT(244, -3950)* *ChkGT(244, -4050*	OK			

3568	Chopper is moved by 4000 (relative). Check that Chopper position is between -50 and 50	*ChkGT(244, -50)*
		ChkLT(244, 50)
			OK			

3569	After the switch off, check status bits in DMC_CHOP_CTRL_ST	*ChkBit(209, 20, 0)*
		ChkBit(209, 21, 0)
			OK			

3570	Chopper is commanded to -4000 with dither. Check that Chopper position is close to -4000	*Chopper pos is close to -4000 (there has been some dither added)*	OK			
3571	Chopper is moved by 4000 with dither. Check that Chopper position is close to 0.	*Chopper pos is close to 0 (there has been some dither added)*	Yes			
3572	Save files 'a_chop.xls' and 'a_chop_nom.xls' (resp. 'a_chop_red.xls')		Yes			
3573	Switch off DEC/MEC		yes			
			Score : 24/24			

9.3.5.4.2 Chopper degraded mode (CHOP.2)

This test is to verify the availability of the Chopper degraded (open loop) control mode. This test is performed with a test connector (@J17, resp. @J117) that simulate the coil resistances with 100-Ω resistors.

Note, the test connector was not available for this test. Therefore, the test has been conducted with the DM chopper connected. Instead of checking the voltage on the coils, we can check that the position of the chopper varies with the number of coils connected

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3574	Connect the test connector ‘**P17_R_CHOP_COIL_SIM**’ @J17 (resp. J117)					
3575	Switch on DEC/MEC and EGSE (see above)					
3576	Execute test script **a_chop_coil.txt**, perform actions as prompted by the test script					
3577	(I = 16mA) When requested, measure voltage at pins 12-13 of the test connector	*4.8V*	Pos = 18812			
3578	(Coil 1 by-passed) When requested, measure voltage at pins 14-13 of the test connector	*3.2V*	Pos = 14875			
3579	(Coil 3 by-passed) When requested, measure voltage at pins 12-15 of the test connector	*3.2V*	Pos = 14675			
3580	(Coils 1 & 3 by-passed) When requested, measure voltage at pins 14-15 of the test connector	*1.6V*	Pos = 9416			
3581	After switch off, when requested, connect the test connector ‘**P17_CHOP_FP_SIM**’ @J17 (resp. J117)					
3582	When requested, measure voltage at pins 21-22 of the test connector	*317mV*	Not tested			
3583	Save files ‘a_chop_coil.xls’ ‘a_chop_coil_nom.xls’ (resp. a_chop_coil_red.xls’					
3584	Switch off DEC/MEC					
3585	Disconnect test connector and reconnect the Chopper as shown on Figure 6-1					
			OK			

9.3.5.4.3 Chopper Short Functional Test (CHOP.3)

This test shows a nominal operation of the chopper

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3586	Switch on DEC/MEC and EGSE (see above)					
3587	Execute test script **func_chop.txt**, perform actions as prompted by the test script		OK			
3588	(This sequence is automatic.)					

9.3.5.5 Calibration Source control

9.3.5.5.1 Calibration source controller (BB.1)

This test is to verify the availability and performance of the calibration sources control functions.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3589	(a) Cal Sources in Spectroscopy mode Disconnect the Calibration Source 1 from J15 and connect the test resistor '**P15_CAL_SRC_ SIM**' (100ohms resistors).		OK			
3590	Execute test script **a_bb2.txt**, perform actions as prompted by the test script		OK			
3591	Set target resistor values to 0ohms		Ok			
3592	Check the resistor measured is 100ohms		Ok			
3593	Set target resistor values to 150 ohms		Ok			
3594	Check the resistor measured is 100ohms		Ok			
3595	In 'hk.xls', check that DMC_CS1&2_OUTPUT is either 0 or +/-327 when in 'measure-only' mode and that the output is bigger when in 'heating-mode'		Ok			
3596	(b) Cal Sources in Photometry mode					
3597	Set target resistor values to 0ohms		Ok			
3598	Check the resistor measured is 100ohms		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3599	Set target resistor values to 150 ohms		Ok			
35100	Check the resistor measured is 100ohms		Ok			
35101	In 'hk.xls', check that DMC_CS1&2_OUTPUT is either 0 or +/-327 when in 'measure-only' mode and that the output is bigger when in 'heating-mode'		Ok			
35102	Save file 'a_bb_nom.xls' (resp a_bb_red.xls)		Ok			
35103	Switch off DEC/MEC					
			18/18			

9.3.5.5.2 Calibration sources calibration (BB.2)

This test is not applicable to a 'software only' acceptance test

9.3.5.6 Filter Wheel mechanism control (FW.1)

This test is to verify the availability of the Filter Wheel mechanisms control functions.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
35104	(a) Spectrometer Filter Wheel in Spectroscopy mode Connect the proto wheel to J13 (resp. J113)		ok			
35105	Execute test script **a_fw.txt**, perform actions as prompted by the test script		ok			
35106	Place the wheel in position A		ok			
35107	Try invalid parameters	*DPU shall receive 2 NACK*	ok			

35108	During the movement to position B, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(210, 25, 1)*
		ChkBit(210, 26, 0)
		ChkBit(210, 27, 1)
		ChkBit(210, 28, 0)
			ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
		ChkBit(210, 29, 0)				

35109	When the wheel has reached position B, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(210, 25, 0)*
		ChkBit(210, 26, 0)
		ChkBit(210, 27, 0)
		ChkBit(210, 28, 0)
		ChkBit(210, 29, 1)
			ok			

35110	During the movement to position A, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(210, 25, 1)*
		ChkBit(210, 26, 1)
		ChkBit(210, 27, 0)
		ChkBit(210, 28, 0)
		ChkBit(210, 29, 0)
			ok			

35111	When the wheel has reached position A, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(210, 25, 0)*
		ChkBit(210, 26, 0)
		ChkBit(210, 27, 0)
		ChkBit(210, 28, 1)
		ChkBit(210, 29, 0)
			ok			

35112	Move the wheel in open loop by ½ turn		ok			

35113	After the command, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(210, 28, 0)*
		ChkBit(210, 29, 1)
			ok			

35114	Move the wheel in open loop by -½ turn		ok			
35115	Save file ‘a_fw_spec.xls’ (resp. ‘a_fw_spec_red.xls’)		ok			
35116	(b) Photometer Filter Wheel in Photometry mode When requested, connect the proto wheel to J14 (resp. J114)		ok			
35117	Place the wheel in position A		ok			
35118	Try invalid parameters	*DPU shall receive 2 NACK*	ok			
35119	During the movement to position B, check status bits in DMC_FW_PHOTO_CTRL_ST	*ChkBit(211, 25, 1)*	ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks

		ChkBit(211, 26, 0)
		ChkBit(211, 27, 1)
		ChkBit(211, 28, 0)
		ChkBit(211, 29, 0)				

35120	When the wheel has reached position B, check status bits in DMC_FW_ PHOTO _CTRL_ST	*ChkBit(211, 25, 0)*
		ChkBit(211, 26, 0)
		ChkBit(211, 27, 0)
		ChkBit(211, 28, 0)
		ChkBit(211, 29, 1)
			ok			

35121	During the movement to position A, check status bits in DMC_FW_ PHOTO _CTRL_ST	*ChkBit(211, 25, 1)*
		ChkBit(211, 26, 1)
		ChkBit(211, 27, 0)
		ChkBit(211, 28, 0)
		ChkBit(211, 29, 0)
			ok			

35122	When the wheel has reached position A, check status bits in DMC_FW_SPEC_CTRL_ST	*ChkBit(211, 25, 0)*
		ChkBit(211, 26, 0)
		ChkBit(211, 27, 0)
		ChkBit(211, 28, 1)
		ChkBit(211, 29, 0)
			ok			

35123	Move the wheel in open loop by ½ turn		ok			

35124	After the second command, check status bits in DMC_FW_ PHOTO _CTRL_ST	*ChkBit(211, 28, 0)*
		ChkBit(211, 29, 1)
			ok			

35125	Move the wheel in open loop by -½ turn		ok			
35126	Save file ‘a_fw_phot.xls’ (resp. ‘a_fw_phot_red.xls’)		ok			
35127	After spectrometer wheel command attempt, check that DPU received a NACK		ok			
35128	Check status bit in DMC_FW_SPEC_CTRL	*ChkBit(211, 20, 0)*	ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
35129	Check status bit in DMC_FW_PHOT_CTRL	*ChkBit(210, 20, 0)*	ok			
35130	Save file 'a_fw_nom.xls'		ok			
35131	Switch off DEC/MEC		ok			
			87/87			

9.3.6 Temperature sensors (TS.1)

For the delta OBS validation, resistors have been connected to every temperature sensors and we checked that the reading of the resistor value was correct.

9.3.7 Detector control

9.3.7.1 Photoconducting detectors control

9.3.7.1.1 Heater DECBASE 1/2 (PHD.1 a)

This test is not applicable to a 'software only' acceptance test.

9.3.7.1.2 Flasher DECBASE 1/2 (PHD.1 b)

This test is not applicable to a 'software only' acceptance test.

9.3.7.1.3 Heater DECBASE 3/4 (PHD.2 a)

This test is not applicable to a 'software only' acceptance test.

9.3.7.1.4 Flasher DECBASE 3/4 (PHD.2 b)

This test is not applicable to a 'software only' acceptance test.

9.3.7.2 Photoconducting arrays acquisition and SPU interface

9.3.7.2.1 Photoconducting arrays acquisition and transfer to blue SPU (PHA.1)

This test is to verify that DMC formats the science data correctly and is able to send it to blue SPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3701	Connect the SPU sim cable to J02					
3702	Execute test script **a_dec2spu.txt**, perform actions as prompted by the test script		Ok			
3703	Start the Link Receiver to replace the Blue SPU, configure it to receive packets and throw them [option 5]		Ok			
3704	The Script will • switch on blue DEC • Start the link with blue SPU • Forward the science data from Blue DEC to SPU		Ok			
3705	In hk.xls, check that DMC_DECB_REC_PAC and DMC_BLUE_ENC_PAC are incrementing by 512 between 2 HK packets		ok			
3706	The Script will • stop forwarding science data from Blue DEC to SPU		ok			
3707	Stop the Link Receiver and restart it, configure it to receive 10 packets in circular buffering mode [option 3]		ok			
3708	The script will • set Blue DEC in simulator mode • Start the link with blue SPU • Forward the science data from Blue DEC to SPU during 2 seconds		ok			
3709	On the SPU Sim, open one of the saved file with an HEX editor and check that the packet is compliant with the SPU-DMC ICD		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3710	The script will • Save the nominal HK • end		ok			
			2/2			

9.3.7.2.2 Photoconducting arrays acquisition and transfer to red SPU (PHA.2)

This test is to verify that DMC formats the science data correctly and is able to send it to red SPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3711	Connect the SPU sim cable to J03		Yes			
3712	Execute test script **a_dec2spu2.txt**, perform actions as prompted by the test script		Yes			
3713	Start the Link Receiver to replace the Red SPU, configure it to save packets on disk [circular numbering with 10 files [option 3]]		Yes			
3714	The script will • Switch on the blue DEC • Set the blue DEC in simulator mode • Start the link with Red SPU • Transfer Blue data to SPU red output • Forward the science data from Blue DEC to SPU		OK			
3715	On the SPU Sim, open one of the saved file with an HEX editor and check that the packet is compliant with the SPU-DMC ICD, especially check the APID		OK			
3716	The script will • stop forwarding science data from Blue DEC to SPU • switch off the blue DEC • Save the nominal HK		OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	• end					
			Score : 1/1			

9.3.7.2.3 Photoconducting arrays simulated data transfer to SPU (PHA.3)

This test is to verify that DMC is able to simulate photo science data and send it to SPU.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3717	Connect the SPU sim cable to J03		Yes			
3718	Execute test script **a_dec2spu3.txt**, perform actions as prompted by the test script		Yes			
3719	Start the Link Receiver to replace the Red SPU, configure it to save packets on disk [circular numbering with 100 files [option 3]]		Yes			
3720	The script will • start link with red SPU • start detector simulator to simulate red DEC at 100Hz • forward science data from red DEC to SPU • stop the detector simulator		OK			
3721	Open one of the saved file with an HEX editor and check that the packet is compliant with the SPU-DMC ICD		OK			
3722	In hk.xls, DMC_DECR_REC_PAC shall increment by 200 between 2 hk packets and stop incrementing after the stop command has been received		OK			
3723	The script will • Save the nominal HK • end		OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
			Score : 2/2			

9.3.7.3 Bolometer arrays acquisition and SPU interface

9.3.7.3.1 Bolometer arrays acquisition and transfer to blue SPU (BOA.1) – Sart a New Test -

This test is to verify that DMC is able to receive science data and hk from BOLC and forward the science data to blue SPU

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3724	Connect the Blue SPU sim cable to J02		Yes			
3725	Restart DEC/MEC		Yes			
3726	Execute test script **a_bol2spu.txt**, perform actions as prompted by the test script		Yes			
3727	The script will • set the timing FPGA in photometry mode		OK			
3728	Start the Link Receiver to replace the BLUE SPU, configure it to receive 256 packets and save them to file [option 4]		OK			
3729	The Script will • Start the link with blue SPU • Reset SMCS chip 2 and connect to BOL • Configure the BOLC to send science data at 40Hz		OK			
3730	In 'Hk.xls', DMC_BOL_REC_PAC shall increment by 480 between 2 hk packets		OK			
3731	The Script will • Forward the data from BOLC to SPU		OK			
3732	Open one of the saved file with an HEX editor and check that the packet is compliant with the SPU-DMC ICD		OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3733	The script will • Save the nominal HK • End		OK			
			Score : 2/2			

9.3.7.3.2 Bolometer simulated data transfer to red SPU (BOA.2)

This test is to verify that DMC is able to receive science data and hk from BOLC and forward the science data to red SPU

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3734	Switch on DEC/MEC and EGSE (see above) Connect SPU sim to J03		Yes			
3735	Execute test script **a_bol2spu2.txt**, perform actions as prompted by the test script		Yes			
3736	Make sure BOLC is switched-off		Yes			
3737	Start the Link Receiver to replace the RED SPU, configure it to receive 256 packets and save them to file [option 4]		OK			
3738	The Script will • start the link with red SPU • start simulating BOLC data at 33Hz • Forward the data from BOLC to SPU		OK			
3739	Open one of the saved file with an HEX editor and check that the packet is compliant with the SPU-DMC ICD		OK			
3740	(leave DEC/MEC on for next test)		Yes			
			Score : 1/1			

9.3.8 Timing

9.3.8.1 With nominal OBT frequency (131,072 Hz) (TIM.1 a)

This test is to verify that DMC can synchronize on OBT and derive its internal timings from it.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3801	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Yes			
3802	Switch on DEC/MEC and EGSE (see above) and make sure that BOLC is ON		Yes			
3803	Execute test script **a_mim1fpga.txt**, perform actions as prompted by the test script		Yes			
3804	Make sure OBT frequency is 131072Hz		OK			
3805	go to spectro mode (timing only)		OK			
3806	set the hk diag list to OBT_COUNT and ISR_COUNT		OK			
3807	switch-on Blue DEC		OK			
3808	connect to BOLC		OK			
3809	synchronize on red DEC		OK			
3810	start HK diag for 1 sec		OK			
3811	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 512 [=131072/256]	*OBT_COUNT increments by 512*	0			No red DEC synchro connected since we are on QM
3812	close hkDiag.xls		OK			
3813	change blue DEC readout frequency to 128Hz		OK			
3814	Synchronize on blue DEC		OK			
3815	start HK diag for 1 sec		OK			
3816	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 1024 [=131072/128]	*OBT_COUNT increments by 1024*	1024			
3817	close hkDiag.xls		OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3818	go to photo mode (timing only). Note, for this test, to have the best accuracy, we use a phase_inc that has been adapted to the BOLC Sim frequency		OK			
3819	configure BOLC to send readouts at 40Hz		OK			
3820	start HK diag for 1 sec		OK			
3821	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3277 or 3276 [=131072/40]	*OBT_COUNT increments by 3276 or 3277*	OK			
3822	close hkDiag.xls		OK			
3823	configure BOLC to send readouts at 2Hz		OK			
3824	start HK diag for 5 sec		OK			
3825	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 65536 [=131072/2]	*OBT_COUNT increments by 65536*	OK			
3826	close hkDiag.xls		OK			
3827	In Hk.xls, The DMC_ISR_COUNT shall increment by 16640		OK			

9.3.8.2 PLL test with nominal OBT frequency (TIM.1 b)

This test is to verify the PLL synchronization.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	(same script as TIM.1)					
3828	Perform actions as prompted by the test script		Yes			
3829	The script changes the BOLC readout frequency and checks the corresponding PLL residue		OK			
3830	BOLC @ 40Hz	*PLL residue high = 16640*	OK			
3831	BOLC @ 20 Hz	*PLL residue high = 33280*	OK			
3832	BOLC @ 10 Hz	*PLL residue high = 1024 (66560 coded on 16bits)*	OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3833	BOLC @ 2 Hz	*PLL residue high = 5120 (332800 coded on 16bits)* *Note that it happens that the PLL residue high = 5119 with a very big value of PLL residue low. This is also an acceptable value*	PLL residue is 5118			The PLL phase inc shall be adapted

Timing parameters are adjusted according to SUM 4.4.20. and test rerun.

9.3.8.3 With non-nominal OBT frequency (130,000 Hz) (TIM.1 c)

This test is to verify that…

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	(same script as TIM.1)					
3834	Change OBT frequency to 130000Hz		Yes			
3835	go to spectro mode (timing only)		OK			
3836	set the hk diag list to OBT_COUNT and ISR_COUNT		OK			
3837	synchronize on blue DEC and change readout frequency to 256Hz		OK			
3838	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 512 [=131072/256]	*OBT_COUNT increments by 512*	OK			
3839	change blue DEC readout frequency to 128Hz		OK			
3840	Synchronize on blue DEC		OK			
3841	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 1024 [=131072/128]	*OBT_COUNT increments by 1024*	OK			
3842	In Hx.xls, DMC_ISR_COUNT shall increment by 16250 [=2*8192*130000/131072]	*DMC_ISR_COUNT shall increment by 16250*	OK			
3843	Don't lock on OBT anymore		OK			
3844	In Hx.xls, DMC_ISR_COUNT shall increment by 16384	*DMC_ISR_COUNT shall increment by 16384*	OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3845	go to photo mode (timing only). Note, for this test, to have the best accuracy, we use a phase_inc that has been adapted to the BOLC Sim frequency		OK			
3846	configure BOLC to send readouts at 40Hz		OK			
3847	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3250 [=130000/40]	*OBT_COUNT increments by 3250*	OK			
3848	configure BOLC to send readouts at 2Hz		OK			
3849	In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 65536 [=130000/2]	*OBT_COUNT increments by 65000*	Increment is 65001 OK			
3850	Make sure OBT frequency is 131072Hz		Score : 18/20			

9.3.9 Interface (INT.1)

This test is to verify that DMC can detect the 1355 disconnections

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3901	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Yes			
3902	Switch on DEC/MEC and EGSE (see above)		Yes			
3903	Execute test script **a_int.txt**, perform actions as prompted by the test script		Yes			
3904	Start a Blue SPU Simulator and configure it to receive packets and throw them [option 5]		OK			

3905	Script will
	• check un-connection/connection of the 1355 link
		ChkBit(202, 19, 1)
		ChkBit(202, 19, 0)
			OK			

3906	disconnect the cable between Blue SPU and DMC [J02 or on the PC side], DMC should detect it and signal the error		OK			
3907	The script will • check that the 1355 is not connected anymore	*ChkBit(202, 19, 1)*	OK			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3908	Nominal HK will be saved	*D:\prj\PACS\TestAcceptance\a_int_nom.xls*	OK			
			Score : 3/3			

9.3.9.1 Synchronization source

This test is a quick test of the synchronization sources

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3801	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Yes			
3802	Switch on DEC/MEC and EGSE (see above) and make sure that BOLC is ON		Yes			
3803	Execute test script **a_synchro.txt**, perform actions as prompted by the test script		Yes			
3804			¾			On DMC EQM, the synchro from the red DEC is not connected

9.3.9.2 Shifted Synchronization source

This test is a quick test of the shifted synchronization source to trigger the mechanism movement

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3805	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		Yes			
3806	Switch on DEC/MEC and EGSE (see above) and make sure that BOLC is ON		Yes			
3807	Execute test script **a_mec_sync.txt**, perform actions as prompted by the test script		Yes			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
3808			1/1			On DMC EQM, the synchro from the red DEC is not connected

9.3.10 Resource (CPU load) (RES.1)

This test is not applicable to a ‘software only’ acceptance test since the QM DMC is not complete.

9.3.11 Reliability (REL.1)

This test is not applicable to a ‘software only’ acceptance test

9.3.12 Trigger commands

This test is to verify all the trigger commands that have not been tested in other tests before

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31201	Verify that the DEC/MEC, EGSE, simulators and instrumentation are connected as per Figure 6-1		ok			
31202	Before the test, Switch off DMC, reconnect Grating to J11, connect a SPU Sim to red SPU link (J03) and switch on DMC		ok			
31203	Execute test script **a_trig1.txt**, perform actions as prompted by the test script		ok			
31204	The script will • Upload the sequence "testLabel" • switch on blue DEC • Synchronize on blue DEC		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	• start the sequence					
31205	In Hk.xls, check that DMC_SEQ_LABEL is changing from 1 to 2 and back in the inner loop.		ok			
31206	The script will • abort the sequence • Write and set new time &check the time has been changed • Write a new time & check the time has not changed yet • set the time • Set a new OBSID & check it has changed • Set a new OBSID & check it has changed • Set a new BBID & check it has changed • Set a new OBSID & check it has changed		ok			
31207	Make sure BOLC sim is connected and switched on		ok			
31208	Script will • Reset SMCS chip 2 and connect to BOLC • Configure the BOLC to send science data • Upload the sequence testWait.seq • synchronize on the Blue DEC • start the sequence and execute it for 10 seconds • Synchronize on the BOLC • start the sequence and execute it for 10 seconds		ok			
31209	In 'Hk.xls', you should see DMC_SEQ_WAIT_IND incrementing by 64 between 2 hk packets during the first execution of the sequence and by 80 during the second execution		ok			
31210	The script will		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	• Change the IRQ frequency • write the new timing parameters in the timing FPGA					
31211	In 'HK.xls' DMC_IRS_CNT shall increment by 8192 between 2 hk packets.		ok			
31212	The script will • switch-on the blue DEC & check that in DMC_DECB_CTRL_ST, bit19 should be 0 (link connected) and bit20=1 (powered on) • Switch-off the blue DEC & check that in DMC_DECB_CTRL_ST, bit19 should be 1 (link disconnected) and bit20=0 (powered off) • switch-on the blue DEC & check that in DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should be 0 • Switch-on blue spectro array & check that In DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 0 to 1 • Switch-off blue spectro array & check that in DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 1 to 0 • Change the number of clocks per readout (64) • Change the number of clocks per readout (32)		ok			
31213	In 'Hk.xls', after the parameters have been changed for the first time DMC_DECB_REC_PAC should increment by 256 between 2 hk packets. After the second command, it should increment by 512.		Ok			
31214	The script will • Switch-off blue spectro array • switch-on the Red DEC & check that in DMC_DECR_CTRL_ST, bit19 should be 0 (link		no			Spectro array of Red DEC can not be switched on on DMC QM.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	connected) and bit20=1 (powered on) • Switch-off the Red DEC & check that in DMC_DECR_CTRL_ST, bit19 should be 1 (link disconnected) and bit20=0 (powered off) • switch-on the Red DEC & check that in DMC_DECR_CR_ST_1 and DMC_DECR_CR_ST_2, bit15 should be 0 • Switch-on Red spectro array & check that In DMC_DECR_CR_ST_1 and DMC_DECR_CR_ST_2, bit15 should change from 0 to 1 • Switch-off Red spectro array & check that in DMC_DECR_CR_ST_1 and DMC_DECR_CR_ST_2, bit15 should change from 1 to 0 • Change the number of clocks per readout (64) • Change the number of clocks per readout (32)					
31215	In 'Hk.xls', after the parameters have been changed for the first time DMC_DECB_REC_PAC should increment by 256 between 2 hk packets. After the second command, it should increment by 512.		Ok			
31216	• switch-on red DEC • send the parameters to both spectro array at the same time		Ok			
31217	In 'Hk.xls', check DMC_DECB_RO_CO_3 and DMC_DECR_RO_CO_1. After the command, they should always have the same value		No			Inapplicable to DMC QM since the red DEC does not get its master clock from the DMC
31218	On the Blue SPU Sim, start 'Blue Link Receiver ' configured to show if the science packets are valid or invalid [option 11].		Ok			
31219	Switch-on BOLC Sim		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31220	The script will • start link with blue SPU and forward blue data from blue DEC to blue SPU	*The Link Receiver shall display 'I'*	Ok			
31221	The script will • validate blue science data	*The Link Receiver shall display 'V'*	Ok			
31222	The script will • invalidate blue science data	*The Link Receiver shall display 'I'*	Ok			
31223	The script will • validate all science data	*The Link Receiver shall display 'V'*	Ok			
31224	The script will • invalidate all science data	*The Link Receiver shall display 'I'*	Ok			
31225	The script will • configure packet encoders such that red SPU data from red DEC goes to blue SPU	*The Link Receiver shall display 'I'*	Ok			
31226	The script will • validate red science data	*The Link Receiver shall display 'V'*	Ok			
31227	The script will • invalidate red science data	*The Link Receiver shall display 'I'*	Ok			
31228	The script will • validate all science data	*The Link Receiver shall display 'V'*	Ok			
31229	The script will • invalidate all science data	*The Link Receiver shall display 'I'*	Ok			
31230	The script will • Reset the BOLC readout counter		NO			This HK does not exist anymore
31231	Check that in 'Hk.xls', DMC_BOL_READ_CNT shall have been reset.		Ok			
31232	The script will		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	• copy the OBS to EEPROM					
31233	In HK.xls , in DMC_SW_GLOBAL_ST, check that bit18=1 during the copy in EEPROM.		Ok			
31234	Switch off DEC/MEC		OK			
			43/48			

9.3.13 Write commands

9.3.13.1 Changing mechanisms parameters (WRT.1)

This test is to verify that mechanisms parameters can be changed using Write commands.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31301	Verify that the DEC/MEC, EGSE, simulators (including Cal Source simulator P1(CAL SRC SIM) and instrumentation are connected as per Figure 6-1		ok			
31302	Switch on DEC/MEC and EGSE (see above)		ok			
31303	Execute test script **a_write1.txt**, perform actions as prompted by the test script		ok			
31304	When requested, check that the value of DMC_CUSTOM_ENT_1 has changed, meaning it is now monitoring another value		ok			
31305	The script will… • Switch on the blue DEC and synchronize on it • Switch on Grating • Set default grating parameters • Enable Grating controller • Home the Grating Wait for the homing completion		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31306	The script will… • Move Grating to central position • Move Grating • Change the rate (3 → 12) • Move grating Wait for the movement completion		ok			
31307	The script will… • Disable Grating controller • Switch off Grating Check in 'hk.xls', DMC_GRAT_SETPOIN is incrementing 4 times faster during the last move , after we have changed the Rate from 3 to 12		ok			
31308	The script will… • Switch on Chopper • Enable Chopper controller • Change hk diag list and start diag hak at 1 kHz • Write default values • Move Chopper • Move Chopper back • Change the rate (keeping default parameters) • Move Chopper • Move Chopper back • Stop hk diag • Disable controller • Switch off Chopper Check 'hkDiag.xls', DMC_CHOP_SETPOIN increments very fast before the write commands, and very slowly after		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31309	Connect the FW @ J13		ok			
31310	The script will… • Switch on FW (spectro) • Write default rate [Write1 (17, 0x64)] • Make complete turn Press OK when first turn is completed		ok			
31311	The script will… • Change the rate [Write1 (17, 0x32)] • Make complete turn Press OK when second turn is completed		ok			
31312	The script will… • Switch FW (spectro) Check in 'hk.xls', in DMC_FW_SPEC_CTRL, you should see in bit25 [moving] that the FW is moving 2 times faster		ok			
31313	Connect the FW @ J14		ok			
31314	The script will… • Switch on FW (photo) • Write default rate [Write1 (18, 0x64)] • Make complete turn Press OK when first turn is completed		ok			
31315	The script will… • Change the rate [Write1 (18, 0x32)] • Make complete turn Press OK when second turn is completed		ok			
31316	The script will… • Switch off FW (photo) Check in 'hk.xls', in DMC_FW_PHOTO_CTRL, you		ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	should see in bit25 [moving] that the FW is moving 2 times faster					
31317	Make sure the Cal Source simulator is connected to BB1		ok			
31318	The script will… • Switch on BB1 and enable controller • Set the target resistor value to 200 Ohms Check that the DMC_CS1_OUTPUT is ±32767		ok			
31319	The script will… • Change the Output limit parameter to 16383 Check that the DMC_CS1_OUTPUT is ±16383		ok			
31320	The script will… • Set back default parameters • Disable BB1 and switch off • Switch on BB2 and enable controller • Set the target resistor value to 200 Ohms Check that the DMC_CS2_OUTPUT is ±32767		ok			
31321	The script will… • Change the Output limit parameter to 16383 Check that the DMC_CS2_OUTPUT is ±16383		ok			
31322	The script will… • Set back default parameters • Disable BB2 and switch off • Change the Max Dither value Check that DMC_CHOP_MAX_DIT goes to value 32		ok			
31323	Save file ‘a_write1_nom.xls’ (resp. ‘a_write1_red.xls’)		OK			
			11/11			

9.3.13.2 Changing detectors parameters (WRT.2)

This test is to verify that detectors parameters can be changed using Write commands.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31324	Execute test script **a_write2.txt**, perform actions as prompted by the test script		OK	OK		
31325	The script will… • Switch to spectroscopy mode (timing only) • Switch on red DEC • Write default parameters for red DEC • Check that number of readouts/ramp = 8 • Write new value (16) • Check that number of readouts/ramp still = 8 • Send parameters to red DEC • Check that number of readouts/ramp = 16 • Switch off red DEC • Switch on blue DEC • Write default parameters for blue DEC • Check that number of readouts/ramp = 8 • Write new value (16) • Check that number of readouts/ramp still = 8 • Send parameters to blue DEC • Check that number of readouts/ramp = 16		ok	ok		
326ok	(go on with next test)					

9.3.13.3 Changing the grating position sensor parameters

This test is to verify the capability of adjusting the inductosyn parameters remotely.

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31327	(Same script as §.9.3.13.2) The script will • switch-on grating • set the default value for the inductosyn amplitude • start hk diag to monitor the sine and cosine amplitude (DMC_GR_IND_SINE and DMC_GR_IND_COS)		ok			
31328	The script will • change the amplitude to 50% of the previous one					
31329	In HkDiag.xls, Maximum values should have decreased by 50%		Ok			
31330	The Script will • set the default value for the inductosyn amplitude and wait until it reaches the value • home toward positive position • disable grating and change the range home toward positive position		Ok			
31331	Check that the mechanical position after the second homing is not the same as after the first homing [there should be a difference of 4 periods of inductosyn], the numerical position should be the same		Ok			
31332	The Script will • Start one homing in the other direction to record		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
	more data • stop hk diag and save diag file for later analysis • configure hk diag list to monitor the position, the setpoint, the output and the ISR counter • start the viewer • start hk diag at 1KHz • change the offset. • Move the grating. Since the hall sensors have a completely wrong value, it should not move correctly					
31333	Press 'OK' if the grating did not move correctly		Ok			
31334	The Script will • write default hall sensor offset again • disable grating controller • enter grating open loop mode • move the grating • move the grating relative • change the rate • move the grating relative • switch off and disable grating		Ok			
31335	Check that The grating should have moved 2 times faster during the second move		Ok			
31336	The Script will • switch-off blue DEC		Ok			

Step	Activity Description	Expected Outcome	Actual Result (N)	Actual Result (R)	Conductor	Remarks
31337	Switch off DEC/MEC		OK			
			11/11			

9.3.14 CRE interface

These tests are not applicable to a 'software only' acceptance test

9.3.15 SPU analog housekeeping (SPUHK.1)

This test is not applicable to a 'software only' acceptance test

9.4 Post-test Activities

This is not applicable to a ‘software only’ acceptance test

10 References

10.1 NCR form

Centre Spatial de Liège	Project : **HERSCHEL - PACS**	NCR N°: **PACS-CL-NCR-000** Rev. **1** Internal N°: Critical Item Yes □ No ☒ Page 1 of 2 Attachments:

Nonconformance Report

NCR title **[title]**			
NC item identification	S/N	Drawing N°	
Next higher assembly		Procedure N°	
Subsystem	Model	Supplier	P.O.
NC observation Date:	Location:	NC detected during…	

Description of nonconformance:	Requirements violated
	Initiator: Date & signature:

Internal NRB dispositions		Ref. to MoMs:		Classification: **Minor** □ **Major** □
				Customer notification per
Cause of NC: Ref. to failure report:		Corrective or preventive actions:		Verification
Date: Name: Signature:	PA	Engineering		
Customer NRB dispositions (Class major, only)		Ref. to MoMs:		Verification
Finally determined cause of NC: Ref. to failure report:		Corrective or preventive actions:		
Request for waiver No ☒ Yes □ Ref. ______________________		Alert No ☒ Yes □ Ref. ______________________		Other related documents:

NRB approval Organisation Name	Chairman					NCR close-out Yes □ No ☒
Date, signature						Date, signature, stamp

Centre Spatial de Liège	Project : HERSCHEL - PACS	NCR N°: **PACS-CL-NCR-000** Rev. **1** Page 2 of 2

Nonconformance Report

- Continutation sheet -

NCR treatment sequence, findings, statements or actions	Verification

10.2 Log book sheet (example)

#	Date & Time	Operator	Observations	References

10.3 TRR check-list

A typical TRR checklist is given below:

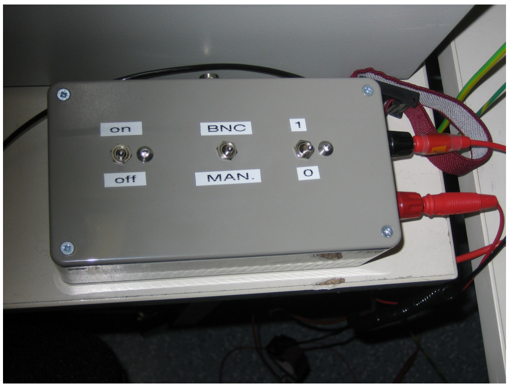
- test objectives and criteria
- test procedure status
- specimen configuration
- GSE status
- facility status
- review of supporting documents
- test personnel status
- safety (of personnel and equipment)
- NCR status
- open work (until test starts)
- planned schedule
- daily and final reporting

10.4 PTR check-list

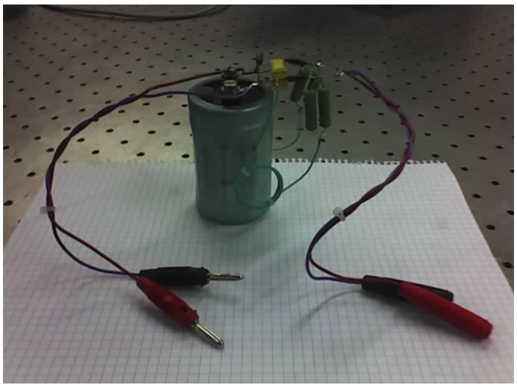
A typical PTR checklist is given below:

- test chronological review (from log-books)
- specimen status
- GSE status
- facility status
- review of procedure changes, anomalies, non-conformances…
- review of test records (pressures, temperatures, cleanliness…)
- review of tested performances
- open work and schedule to completion
- reporting requirements

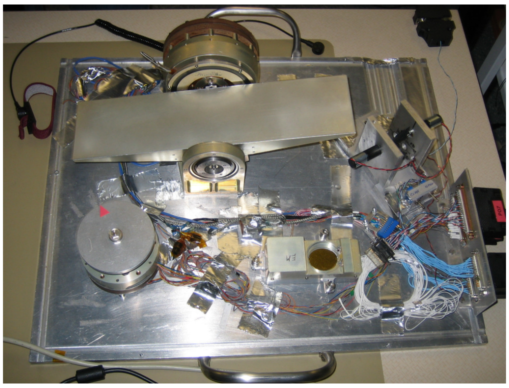
Annex 1 – GSE Descriptions



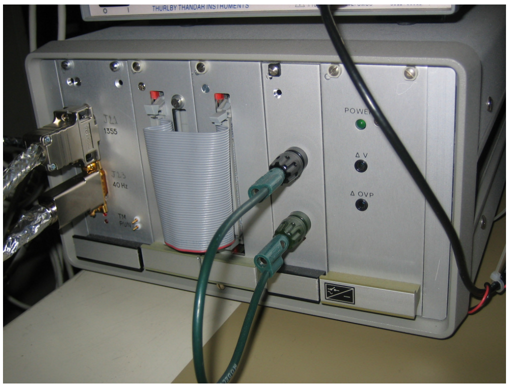
MOSFET Switch



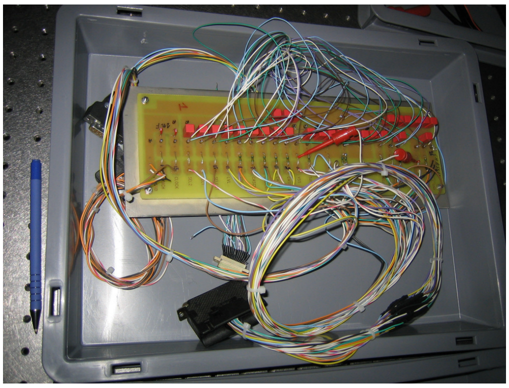
LISN



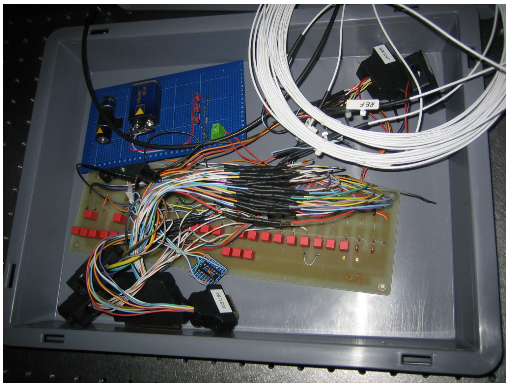
FPU Mechanism Simulators



BOLC Simulator (courtesy of CEA)

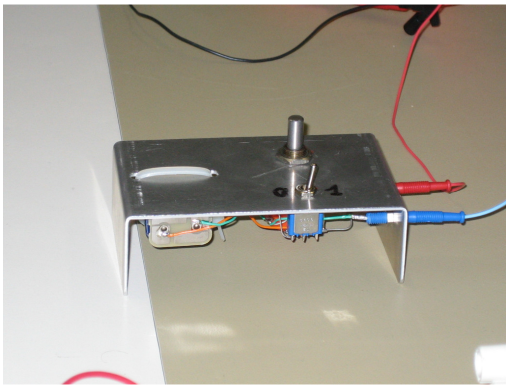


Distribution Board Simulator (1)



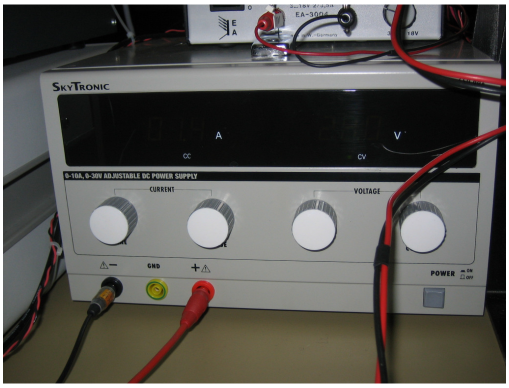
Distribution Board Simulator (2)

Temperature Sensor Simulators



TBD

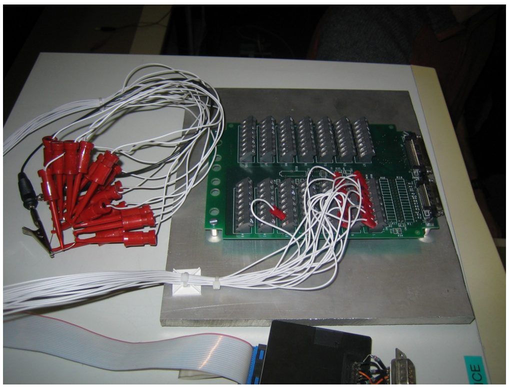
Annex 2 – Instrumentation Descriptions



Lab Power Supply

Frequency Generator

Digital Oscilloscope & Current Amplifier



Acquisition Board

Digital Multi-meter (DATRON)

Annex 3 – Test Scripts

Test script language:

- # defines the comments
- **Trig(ID, nbParams, Param)** sends a trigger command to DMC
    - Example: Trig(12, 0, 0) switches on blue DEC (it is a command with zero parameter)
    - Trig(76, 1, 5) starts the diagnostic hk with a period of 5ms
    - You will find a list of trigger commands in the DMC SUM. As often as possible, a comment in the code will tell you what it does
- **WriteN(ID, param1, param2, …)** sends a write command to DMC
    - N is the number of parameter
    - ID is the write command identifier
    - Example: Write3(13, 250, 251, 0xFFFF) defines the list of diag hk.
    - You will find a list of write commands in the DMC SUM. As often as possible, a comment in the code will tell you what it does
- **WaitTime(t, message)** wait for t seconds and eventually displays a message. We often use it to wait for the next hk packets before checking the content of the hk packet
- **WaitForGo(message)** requests the operator to perform an external operation (connect something, switch-on something, …). Once the operator has clicked OK, the script execution continues.
- **Repeat(n) – EndRepeat()** are used to perform loops
- **VarInt(varName)** declares a variable named 'varName'
- **SetInt(varName, val)** set the value of varName to val
- **IncInt(varName, inc)** increments the value of varName by inc
- **ManualCheck(message)** requests the user to check something that can not be checked in the HK (measure something with a voltmeter, …)
- **ChkEQ(HK_ID, value)** checks that the HK_ID measure is equal to value
- **ChkGT(HK_ID, value)** checks that the HK_ID measure is greater then value
- **ChkLT(HK_ID, value)** checks that the HK_ID measure is lower then value
- **ChkBit(HK_ID, bitPos, bitValue)** checks that the bit bitPos in the HK_ID measure is set to bitValue
- **PrintHk(HK_ID)** prints the value of HK_ID measure
- **ChkClearReport()** clears the test counters. Each time a check is performed, counters are incremented to be able to display the test results at the end of the script
- **ChkReport()** displays the test results at the end of the script
- **DmcIsAlive()** is used to tell the DPU that DMC is still alive. This must be done after DPU has received a NACK from DMC.
- **UploadSeq(filename)** upload a sequence in DMC
- **Log(message)** displays a message on the screen
- **System(command)** executes a system command

SWON.1: Switch-on
Script file: a_swon.txt

During the power-on, look at the mechanisms. Make sure they are all connected before starting the test

```
ChkClrReport()

#at the power on, the mechanisms shall not move
ManualCheck("Click YES if the mechanisms did not move at the power on");

#check that DMC is connected to DPU Sim
ChkBit(198, 19, 0)
ChkBit(199, 19, 0)

#switch-on blue DEC, red DEC, chopper and grating
Trig(12, 0, 0)
Trig(19, 0, 0)
Trig(49, 0, 0)
Trig(38, 0, 0)
WaitTime(7, "")

#check that blue DEC, red DEC, chopper and grating are powered-on
ChkBit(201, 20, 1)
ChkBit(204, 20, 1)
ChkBit(209, 20, 1)
ChkBit(208, 20, 1)

ChkReport()
```

SWOF.1: Switch-off
Script file: a_swof.txt

Restart a new test (switch-off and on DMC)

```
ChkClrReport()

#after a fresh power-on of DMC, all sub-systems shall be in their default state (off)
#check that blue DEC, chopper and grating are powered-off
ChkBit(201, 20, 0)
ChkBit(204, 20, 0)
ChkBit(209, 20, 0)
ChkBit(208, 20, 0)

ChkReport()
```

HKN.1: Nominal housekeeping
Script file: a_hk.txt

```
ChkClrReport()

#check that hk is received by DPU (if yes, it is stored in the file hk.dat)
ManualCheck("Check in ‘d:\prj\pacs\simulators\simdpuv26\asw’ that a file hk.dat has been
created and that it is updated regularly")

System("D:\prj\PACS\TestPlanTools\View_HK_as_text.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.txt")
ManualCheck("in ‘Hk.txt’ contains the whole set of  housekeeping variables.")

ChkReport()
```

HKD.1: Test diagnostic housekeeping
Script file: a_hkDiag.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#delete old hk diag file if any
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

#set the hk diag list to DMC_DECB_REC_PAC + DMC_IRS_CNT
Write3(13, 228, 242, 0xFFFF)

#start hk diag at 200Hz
Trig(76, 1, 5)
ManualCheck("Check that d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat is updated")

System("D:\prj\PACS\TestPlanTools\View_HK_diag_HKD.2.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("check that HkDiag.xls  contain the 2 variables sampled at 200Hz")

#stop hk diag and delete file
Trig(77, 0, 0)
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

#switch on blue dec and wait that it's on
Trig(12, 0, 0)
WaitTime(5, "")

#start hk diag synchronized on blue DEC
Trig(76, 1, 1)
WaitTime(3, "")

#stop hk diag
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_HKD.2.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("in HkDiag.xls check that DMC_DECB_REC_PAC increment by one at each sample")

#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_hkDiag_nom.xls")
ChkReport()
```

DPU.1: DPU Interface
Script file: a_dpu.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#synchronize on blue DEC (this is a valid command)
Trig(10, 1, 1)
ManualCheck("Did DPU received a PACK ?")

#same command witht the wrong number of parameters
Trig(10, 0, 0)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_dpu_nom.xls")
ChkReport()
```

IC.1: Sequences
Script file: a_seq.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#upload sequence to DMC
# DMC_LOOP , 5
#   DMC_LOOP , 2
#     DMC_WAIT , 160
#   DMC_END_LOOP , 0
# DMC_END_LOOP , 0
# DMC_END_SEQUENCE , 0
UploadSeq("loop_wait")

#switch on blue DEC
Trig(12, 0, 0)
WaitTime(5, "")

#synchronize on blue DEC
Trig(10, 1, 1)
WaitTime(0.5, "")

#start the sequence execution
Trig(5, 0, 0)
WaitTime(30, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In 'Hk.xls', you should see that: DMC_SEQ_LOOP_ID0 is decreasing from 5 to 1
DMC_SEQ_LOOP_ID1 is decreasing from 2 to 1 5 times. DMC_SEQ_WAIT_IND is increasing from 0
to 160 10 times.")

#abort sequence execution
Trig(6, 0, 0)

#switch on BOLC and connect the 1355 link
WaitForGo("Switch on BOLC if not already done")
Trig(89, 0, 0)
WaitTime(1, "")

#configure BOLC to send readouts
Trig(33, 1, 0x09020002)

#syncrhonize on BOLC
Trig(10, 1, 4)
WaitTime(0.5, "")

#start the sequence execution
Trig(5, 0, 0)
WaitTime(30, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In 'Hk.xls', you should see that: DMC_SEQ_LOOP_ID0 is decreasing from 5 to 1
DMC_SEQ_LOOP_ID1 is decreasing from 2 to 1 5 times. DMC_SEQ_WAIT_IND is increasing from 0
to 160 10 times. Note that now that we are synchronized on BOLC, DMC_WAIT_IND should
increase faster")

#show that the only command that is accepted during a sequence execution is the
DMC_ABORT_SEQUENCE
#abort the sequence just to make sure it is not running when starting this test
Trig(6, 0, 0)
Wait(0.5, "")

#start the sequence and abort it 3 seconds later
Trig(5, 0, 0)
Wait(3, "")
Trig(6, 0, 0)
ManualCheck("Did DPU received a PACK ?")

#start the sequence again and try to send another trigger command (to change the synchro
source)
Trig(5, 0, 0)
Wait(3, "")
Trig(10, 1, 1)
```

```
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#show that write commands are rejected during a sequence execution
#abort the sequence just to make sure it is not running when starting this test
Trig(6, 0, 0)
Wait(0.5, "")

#start the sequence execution
Trig(5, 0, 0)
WaitTime(3, "")

#try to send a write command (to change SPU transmission mode)
Write2(28, 0, 0)
ManualCheck("Did DPU received a NACK ?")
WaitTime(3, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, DMC_B_SPU_TR_MODE shall not have been modified ?")
DmcIsAlive()

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_seq_nom.xls")
ChkReport()
```

ICM.1: Mechanisms control
Script file: a_mec.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#check that the mechanism control interrupt service routine is running
WaitTime(10, "")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, Is DMC_IRS_CNT incrementing by 16384 ?")

#show that the mechanisms movement are synchronized with DEC synchro
#switch off blue DEC
Trig(13, 0, 0)
WaitTime(0.5, "")

#select the blue DEC as the synchronization source (without using synchro for mechanisms
move)
Trig(10, 1, 9)
WaitTime(0.5, "")

#switch-on and enable chopper
Trig(49, 0, 0)
WaitTime(0.5, "")
Trig(51, 0, 0)
WaitTime(3, "")

#make sure that the chopper position is close to zero
ChkGT(244, -1000)
ChkLT(244, 1000)

#move the chopper.  It should move directly since it does not require a synchro source
Trig(53, 1, 6000)
WaitTime(3, "")
ChkGT(244, 5000)
ChkLT(244, 7000)
Trig(53, 1, 0)
WaitTime(0.5, "")

#select the blue DEC as the synchronization source (using synchro for mechanisms move)
Trig(10, 1, 1)
WaitTime(0.5, "")

#try to move chopper (since the source of synchro is powered off, it shall not move)
Trig(53, 1, 6000)
WaitTime(3, "")
ChkGT(244, -1000)
```

```
ChkLT(244, 1000)

#now, switch-on blue DEC (the chopper shall move now since it is still waiting for the
synchro and the synchro will come now)
Trig(12, 0, 0)
WaitTime(7, "")
ChkGT(244, 5000)
ChkLT(244, 7000)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_mec_nom.xls")
ChkReport()
```

GRAT.1: Grating launch lock
Script file: a_grat_ll.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Make sure that both nominal and redundant are connected to DMC")

#start hk diag at 200Hz to record LL current and Grating status
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
Write4(13, 546, 570, 208, 0xffff)
Trig(76, 1, 5)

WaitForGo("On the LL simulator, put the 'close' switch ON [down] and the 'open' switch OFF
[up]")
#switch on grating
Trig(38, 0, 0)
WaitTime(3, "")

#check LL status (LL not moving and both switches are at 0 when LL is not moving)
ChkBit(208, 19, 0)
ChkBit(208, 30, 0)
ChkBit(208, 31, 0)

#send the unlock command
Trig(48, 1, 0x28)
WaitTime(3, "")

#check LL status (LL moving and switches telling it is 'locked')
ChkBit(208, 19, 1)
ChkBit(208, 30, 1)
ChkBit(208, 31, 0)

WaitForGo("On the LL simulator, put the 'close' switch OFF [up] and the 'open' switch ON
[down]")
WaitTime(3, "")
ChkBit(208, 19, 0)

#send the lock command
Trig(47, 1, 0x12)
WaitTime(3, "")

#check LL status (LL moving and switches telling it is 'unlocked')
ChkBit(208, 19, 1)
ChkBit(208, 30, 0)
ChkBit(208, 31, 1)

WaitForGo("On the LL simulator, put the 'close' switch OFF [up] and the 'open' switch OFF
[up]")
WaitTime(3, "")
ChkBit(208, 19, 0)

#send the lock command
Trig(47, 1, 0x12)
WaitTime(3, "")

#check LL status (LL moving and switches neither locked nor unlocked)
ChkBit(208, 19, 1)
ChkBit(208, 30, 0)
ChkBit(208, 31, 0)
```

```
#check commanding the actuators separately
WaitForGo("On the LL simulator, disconnect actuator 2 [connected to redundant]")
Trig(48, 1, 0x08)
WaitTime(6, "")
Trig(47, 1, 0x02)
ManualCheck("Did the LL moved twice in opposite directions ?")

Trig(48, 1, 0x20)
WaitTime(6, "")
Trig(47, 1, 0x10)
ManualCheck("The LL should not have moved")

WaitForGo("On the LL simulator, reconnect actuator 2 [connected to redundant]")
Trig(48, 1, 0x20)
WaitTime(6, "")
Trig(47, 1, 0x10)
ManualCheck("Did the LL moved twice in opposite directions ?")

Trig(47, 1, 0x12)
ManualCheck("The LL should have moved a little faster")

Trig(39, 0, 0)

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_grat_ll.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_grat_ll.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_grat_ll_nom.xls")
ChkReport()
```

GRAT.2: Grating closed loop
Script file: a_grat.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#Make sure we can execute the script: switch-off grating, stop hk diag if it was running
Trig(39, 0, 0)
Trig(77, 0, 0)


#Write STM Grating params
#Write Hall sensors offset
Write1(34, 636);
#Write Range
Write1(33, 0x100000)
#Nominal parameters
Write8(15, 0x1388, 0x3d090, 0x28, 0, 0x3, 0x10e4311, 0x452f, 1631)

#switch on grating and enable the controller
Trig(38, 0, 0)
WaitTime(0.5, "")
Trig(40, 0, 0)
WaitTime(2.5, "")

#check that the homing has not been performed yet and that it is not running now
ChkBit(208, 28, 0)
ChkBit(208, 29, 0)

#start a homing with a wrong parameter
Trig(44, 1, 2)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()
Trig(44, 1, -1)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#start a hk diag at 1KHz with DMC_GRAT_CUR_POS, DMC_GRAT_SETPOIN, DMC_FW_GR_VMOTA,
#DMC_FW_GR_VMOTB, DMC_FW_GR_IMOTA, DMC_FW_GR_IMOTB, DMC_FWGRAT_HALLA, DMC_FWGRAT_HALLB,
#DMC_GRAT_OUTPUT for further analysis by CSL
```

```
Write10(13, 250, 251, 556, 560, 564, 567, 256, 257, 452, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 0)

#start a homing
Trig(44, 1, 0)
WaitTime(3, "")

#check that the homing is in progress
ChkBit(208, 28, 1)
WaitForGo("Wait Homing is completed")

#check that the homing has been performed and that it is not running now
ChkBit(208, 28, 0)
ChkBit(208, 29, 1)

#move the grating absolute
Trig(42, 1, 0x40000)
WaitTime (15,"")

#check that the grating has reached the expected position
ChkGT(250, 0x3FFF0)
ChkLT(250, 0x40010)

#move the grating relative
Trig(43, 1, 0x10000)
WaitTime (5,"")

#check that the grating has reached the expected position
ChkGT(250, 0x4FFF0)
ChkLT(250, 0x50010)

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_grat_all.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_grat.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

#disable it and switch-it off
Trig(41, 0, 0)
Trig(39, 0, 0)

#check that it has been switched off and that the position is now invalid
WaitTime(5, "")
ChkBit(208, 20, 0)
ChkEQ(250, -1)

#test the limit switch
#switch-on grating but does not enable the controller
Trig(38, 0, 0)
WaitTime(3, "")

#check that the limit switch is not pressed
ChkBit(208, 23, 0)

WaitForGo("Press the grating on the limit switch [FW side]")
WaitTime(3, "")

#check that the limit switch is pressed
ChkBit(208, 23, 1)

WaitForGo("Press OK once you have released the grating");
WaitTime(3, "")

#check that the limit switch is not pressed
ChkBit(208, 23, 0)

#switch-off the grating
Trig(39, 0, 0)


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_grat_nom.xls")
ChkReport()
```

GRAT.3: Grating open loop mode
Script file: a_grat_open.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")
Trig(77, 0, 0)
WaitTime(0.5, "")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")

Write10(13, 250, 251, 556, 560, 564, 567, 256, 257, 452, 0xFFFF)       #write new hk diag
list
Trig(76, 1, 0)

#switch-on and enter closed loop
Trig(38, 0, 0)
WaitTime(0.5, "")
Trig(40, 0, 0)
WaitTime(4, "")

#check in DMC_GRAT_CTRL_ST that it is in closed loop and that it is not in degraded mode
ChkBit(208, 21, 1)
ChkBit(208, 24, 0)

#try to enter open loop mode (it should not be possible now)
Trig(45, 1, 0)
WaitTime(3, "")
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#check in DMC_GRAT_CTRL_ST that it is in closed loop and that it is not in degraded mode
ChkBit(208, 21, 1)
ChkBit(208, 24, 0)

#switch-off
Trig(39, 0, 0)
WaitTime(3, "")

#check in DMC_GRAT_CTRL_ST that it is not in closed loop and that it is not in degraded
mode
ChkBit(208, 21, 0)
ChkBit(208, 24, 0)

#switch-on and enter open loop
Trig(38, 0, 0)
Trig(45, 1, 0)
WaitTime(3, "")

#check in DMC_GRAT_CTRL_ST that it is not in closed loop and that it is in degraded mode
ChkBit(208, 21, 0)
ChkBit(208, 24, 1)

#try to enter closed loop mode (it should not be possible now)
Trig(40, 0, 0)
WaitTime(3, "")
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#check in DMC_GRAT_CTRL_ST that it is not in closed loop and that it is in degraded mode
ChkBit(208, 21, 0)
ChkBit(208, 24, 1)

#now, move the grating in open loop
Trig(42, 1, 0x1000)
WaitTime(20, "")
PrintHk(250)

#move the grating relative
Trig(43, 1, 0x1000)
WaitTime(20, "")
PrintHk(250)

#move the grating relative back to its original position
Trig(43, 1, -0x1000)
WaitTime(20, "")
PrintHk(250)
ManualCheck("Check that the position after the first move is 'close to' the position after
the third move")

#exit degraded mode
Trig(46, 0, 0)
WaitTime(3, "")

#check in DMC_GRAT_CTRL_ST that it is not in closed loop and that it is not in degraded
mode
ChkBit(208, 21, 0)
ChkBit(208, 24, 0)

#switch-off grating
Trig(39, 0, 0)

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_grat_all.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_grat_open.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_grat_open_nom.xls")
ChkReport()
```

GRAT.4: Grating short functional test
Script file: func_grat.txt

```
Log("* Grating functional test")

#upload test sequence to DMC
# DMC_LOOP , 100
#   DMC_MOVE_GRAT_REL , 117
#   DMC_WAIT , 32
# DMC_END_LOOP , 0
# DMC_END_SEQUENCE , 0
UploadSeq("gratSteps")

#Write STM Grating params
#Write Hall sensors offset
Write1(34, 636);
#Write Range
Write1(33, 0x100000)
#Nominal parameters
Write8(15, 0x1388, 0x3d090, 0x28, 0, 0x3, 0x10e4311, 0x452f, 1631)

#switch-on blue DEC
Trig(12, 0, 0)
WaitTime(5, "")

#synchronize on blue DEC
Trig(10, 1, 1)
WaitTime(0.5, "")

#Switch-on grating
Trig(38, 0, 0)
WaitTime(0.5, "")

#configure hk diag list to monitor the position, the setpoint, the output and the ISR
counter
Write5(13, 0xFA, 0xFB, 0x1C4, 0xF2, 0xFFFF)

#start the viewer
WaitForGo("Launch HkDiagRTViewer")

#start hk diag at 1KHz
Trig(76, 1, 0)

#enable grating controller
Trig(40, 0, 0)

#home grating
Trig(44, 1, 1)
WaitForGo("Wait homing is completed")
```

```
#start sequence execution
Trig(5, 0, 0)
WaitForGo("Wait sequence is completed")

#abort the sequence
Trig(6, 0, 0)

#stop hk diag
Trig(77, 0, 0)
WaitTime(0.5, "")

#switch off blue DEC
Trig(13, 0, 0)
WaitTime(0.5, "")

#disable grating and switch-it off
Trig(41, 0, 0)
WaitTime(0.5, "")
Trig(39, 0, 0)
```

CHOP.1: Chopper controller
Script file: a_chop.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#*************************************************
# TEST CHOPPER IN SPECTRO MODE
#*************************************************

#set the timing FPGA in spectrometry mode
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")

#switch-on blue DEC and synchronize on it
Trig(12, 0, 0)
WaitTime(5, "")
Trig(10, 1, 1)

# switch on chopper
Trig(49, 0, 0)
WaitTime(0.5, "")

#start a hk diag at 1KHz with DMC_CHOP_CUR_POS, DMC_CHOP_SETPOIN, DMC_CHOP_OUTPUT,
DMC_CHOP_TARGET,
#DMC_CHOP_PID_ERR, DMC_CHOP_PID_ACC, DMC_CHOP_VA, DMC_CHOP_VB, DMC_CHOP_IA for further
analysis by CSL
Write10(13, 244, 245, 258, 246, 247, 248, 557, 565, 561, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 0)

#set the chopper parameters for DM Tamb
Write21(16, 535080, 33261737, 304, 1114, 293, 0x3FFFFFFF, 0x7fff, 29000, 0x7FFFFFFF, -610,
140000, 0, 0, 101000, 172000, 101000, 1839000, 900000, 137, 652000, 8000)

# enable chopper controller
Trig(51, 0, 0)
WaitTime(3, "")

#After the enable command, check status bits in DMC_CHOP_CTRL_ST it should be powered on
and enabled
ChkBit(209, 20, 1)
ChkBit(209, 21, 1)

#check that the position is around zero
ChkGT(244, -300)
ChkLT(244, 300)

# move chopper to position 4000
Trig(53, 1, 4000)
WaitTime(3, "")
```

```
# check that the chopper position is between 3950 and 4050
ChkGT(244, 3950)
ChkLT(244, 4050)

# move chopper relative to the central position
Trig(54, 1, -4000)
WaitTime(3, "")

# check that the chopper position is between -50 and 50
ChkGT(244, -50)
ChkLT(244, 50)

# move chopper to position 4000 with dither
Trig(55, 1, 4000)
WaitTime(3, "")

PrintHk(244)
ManualCheck("Check that the chopper position is close to 4000 [there has been some dither
added]")

# move chopper relative to the central position with diter
Trig(56, 1, -4000)
WaitTime(3, "")

PrintHk(244)
ManualCheck("Check that the chopper position is close to 0 [there has been some dither
added]")

#disable chopper controller
Trig(52, 0, 0)
WaitTime(0.5, "")

#switch off chopper
Trig(50, 0, 0)
WaitTime(3, "")

#After the switch off, check status bits in DMC_CHOP_CTRL_ST : it should be powered off and
disabled
ChkBit(209, 20, 0)
ChkBit(209, 21, 0)


#*************************************************
# TEST CHOPPER IN PHOTO MODE
#*************************************************

#set the timing FPGA in photometry mode
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#switch-on BOLC and synchronize on it
WaitForGo("Make sure BOLC is switched on")
Trig(89, 0, 0)

# switch on chopper
Trig(49, 0, 0)
WaitTime(0.5, "")

#set the chopper parameters for DM Tamb
Write21(16, 535080, 33261737, 304, 1114, 293, 0x3FFFFFFF, 0x7fff, 29000, 0x7FFFFFFF, -610,
140000, 0, 0, 101000, 172000, 101000, 1839000, 900000, 137, 652000, 8000)

# enable chopper controller
Trig(51, 0, 0)
WaitTime(3, "")

#After the enable command, check status bits in DMC_CHOP_CTRL_ST it should be powered on
and enabled
ChkBit(209, 20, 1)
ChkBit(209, 21, 1)

#check that the position is around zero
```

```
ChkGT(244, -300)
ChkLT(244, 300)

# move chopper to position -4000
Trig(53, 1, -4000)
WaitTime(3, "")

# check that the chopper position is between 3950 and 4050
ChkLT(244, -3950)
ChkGT(244, -4050)

# move chopper relative to the central position
Trig(54, 1, 4000)
WaitTime(3, "")

# check that the chopper position is between -50 and 50
ChkGT(244, -50)
ChkLT(244, 50)

# move chopper to position -4000 with dither
Trig(55, 1, -4000)
WaitTime(3, "")

PrintHk(244)
ManualCheck("Check that the chopper position is close to -4000 [there has been some dither
added]")

# move chopper relative to the central position with diter
Trig(56, 1, 4000)
WaitTime(3, "")

PrintHk(244)
ManualCheck("Check that the chopper position is close to 0 [there has been some dither
added]")

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_chop_all.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_chop.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

#disable chopper controller
Trig(52, 0, 0)
WaitTime(0.5, "")

#switch off chopper
Trig(50, 0, 0)
WaitTime(3, "")

#After the switch off, check status bits in DMC_CHOP_CTRL_ST : it should be powered off and
disabled
ChkBit(209, 20, 0)
ChkBit(209, 21, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_chop_nom.xls")
ChkReport()
```

CHOP.2: Chopper degraded mode
Script file: a_chop_coil.txt

```
Before starting this test:
Switch-off DMC,
Connect P17 R CHOP COIL SIM (this is a chopper simulator where each coil is replaced by a
100ohms resistor)
Switch-on DMC and start OBS.
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

# switch on chopper
Trig(49, 0, 0)
WaitTime(0.5, "")

#start a hk diag at 20Hz with CHOP_VA, CHOP_IA and CHOP_VB for further analysis by CSL
Write4(13, 557, 561, 565, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 50)

#set the controller in open-loop mode and enable it
Trig(57, 1, 0x128)
Trig(51, 0, 0)

#set the current to 16mA
Trig(53, 1, 0x1000)
ManualCheck("check that there is 4.8V between pin12 and pin13")

#set the controller in open-loop mode and bypass coil 1
Trig(52, 0, 0)
Trig(57, 1, 0x130)
Trig(51, 0, 0)
Trig(53, 1, 0x1000)
ManualCheck("check that there is 3.2V between pin14 and pin13")

#set the controller in open-loop mode and bypass coil 3
Trig(52, 0, 0)
Trig(57, 1, 0x148)
Trig(51, 0, 0)
Trig(53, 1, 0x1000)
ManualCheck("check that there is 3.2V between pin12 and pin15")

#set the controller in open-loop mode and bypass coil 1 and 3
Trig(52, 0, 0)
Trig(57, 1, 0x150)
Trig(51, 0, 0)
Trig(53, 1, 0x1000)
ManualCheck("check that there is 1.6V between pin14 and pin15")

#set the controller in open-loop mode with all coils active
Trig(52, 0, 0)
Trig(57, 1, 0x128)
Trig(51, 0, 0)
Trig(53, 1, 0x1000)

#stop hk diag
Trig(77, 0, 0)

#disable controller and switch-off chopper
Trig(52, 0, 0)
Trig(50, 0, 0)

#check the field plate driver voltage
WaitForGo("Connect P17 CHOP FP SIM")
Trig(49, 0, 0)

ManualCheck("Check that the voltage between pin 21 and 22 is 317mV")
Trig(50, 0, 0)

#save the analog hk of the chopper for further analysis
System("D:\prj\PACS\TestPlanTools\View_HK_diag_chop_ampli.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_chop_coil.xls")

#delete old hk diag file
```

```
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_chop_coil_nom.xls")
ChkReport()
```

CHOP.3: Chopper short functional test
Script file: func_chop.txt

```
Before starting this test:
Switch-off DMC,
Reconnect the chopper to DMC
Switch-on DMC and start OBS.
```

```
Log("* Chopper functional test")

# upload the sequence simpleChop.seq
# DMC_LOOP , 100
#   DMC_MOVE_CHOP_ABS , 13434
#   DMC_WAIT , 3
#   DMC_MOVE_CHOP_ABS , -13434
#   DMC_WAIT , 3
# DMC_END_LOOP , 0
# DMC_MOVE_CHOP_ABS , 0
# DMC_END_SEQUENCE , 0
UploadSeq("simplechop")

#set the chopper parameters for DM Tamb
Write21(16, 535080, 33261737, 304, 1114, 293, 0x3FFFFFFF, 0x7fff, 29000, 0x7FFFFFFF, -610,
140000, 0, 0, 101000, 172000, 101000, 1839000, 900000, 137, 652000, 8000)


#switch on the chopper
Trig(49, 0, 0)
WaitTime(0.5, "")

# change the hk diag list
Write5(13, 244, 245, 258 , 242, 0xFFFF)

WaitForGo("Launch HkDiagRTViewer")

# start diag hk at 1Khz
Trig(76, 1, 0)

# enable the chopper controller
Trig(51, 0, 0)

# switch on blue DEC
Trig(12, 0, 0)
WaitTime(5, "")

# synchronize on blue DEC
Trig(10, 1, 1)
WaitTime(0.5, "")

# start the sequence
Trig(5, 0, 0)
WaitForGo("Wait sequence is completed")
Trig(6, 0, 0)

# stop hk diag
Trig(77, 0, 0)
WaitTime(0.5, "")

# switch off blue DEC
Trig(13, 0, 0)
WaitTime(0.5, "")

#disable chopper controller
Trig(52, 0, 0)
WaitTime(0.5, "")

#switch off the chopper
Trig(50, 0, 0)
```

BB.1: Calibration source
Script file: a_bb2.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("For tests with proto: Replace the calibration source 1&2 by 100 ohms resistor")

#switch on BB1 and enable controller
Trig (68,0,0)
Trig (91,0,0)

#switch on BB2 and enable controller
Trig (72,0,0)
Trig (93,0,0)

#start a hk diag at 20Hz
Write11(13, 445, 446, 522, 523, 524, 525, 526, 527, 528, 529, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 50)

############################################
Log("Test Cal Src 1&2 in spectro timing mode")
############################################

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)

#set the target resistor value to 0ohms.
Trig (70, 1, 0)
Trig (74, 1, 0)

#wait to get the first measures
WaitTime (45,"")
ChkGT(445, 990000)
ChkLT(445, 1010000)
ChkGT(447, 990000)
ChkLT(447, 1010000)

#set the target resistor value to 150ohms.
Trig (70, 1, 1500000)
Trig (74, 1, 1500000)
WaitTime (45,"")

ChkGT(445, 990000)
ChkLT(445, 1010000)
ChkGT(447, 990000)
ChkLT(447, 1010000)

#set the calibration sources in "simulation mode"
Trig(90, 1, 0x30)
WaitTime (45,"")
ChkEQ(445, 1500000)
ChkEQ(447, 1500000)

#exit simulation mode
Trig(90, 1, 0x00)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'hk.xls', check that DMC_CS1_OUTPUT is either 0 or +/-327 when in ‘measure
only’ mode and that the output is bigger when in ‘heating mode’.")

############################################
Log("Test Cal Src 1 in photo timing mode")
############################################

#go to photo mode (timing only)
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
```

```
Trig(11, 0, 0)
WaitTime(1, "")

#set the target resistor value to 0ohms.
Trig (70, 1, 0)
Trig (74, 1, 0)

#wait to get the first measures
WaitTime (45,"")
ChkGT(445, 990000)
ChkLT(445, 1010000)
ChkGT(447, 990000)
ChkLT(447, 1010000)

#set the target resistor value to 150ohms.
Trig (70, 1, 1500000)
Trig (74, 1, 1500000)
WaitTime (45,"")

ChkGT(445, 990000)
ChkLT(445, 1010000)
ChkGT(447, 990000)
ChkLT(447, 1010000)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'hk.xls', check that DMC_CS1_OUTPUT is either 0 or +/-327 when in ‘measure
only’ mode and that the output is bigger when in ‘heating mode’.")

#disable BB controllers and switch-off
Trig (92,0,0)
Trig (69,0,0)
Trig (94,0,0)
Trig (73,0,0)

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_CAL_SRC1.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_bb1.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
```

BB.2: Calibration source calibration
Script file: cal_bb.txt

```
WaitForGo ("For tests with proto: Replace the calibration source 1 by a variable resistor.
Connect P15 CAL SRC VAR SIM")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#2-68-0 to switch on BB1
Trig (68,0,0)
WaitForGo ("Set the resistor value to 30 ohms")
WaitTime (24,"")
PrintHk(445)

#set the voltage to 2V
Trig(71, 1, 6553)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 3V
Trig(71, 1, 9830)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 4V
Trig(71, 1, 13106)
WaitTime (24,"")
PrintHk(445)

#back to measure only mode
Trig(71, 1, 0)
WaitForGo ("Set the resistor value to 60 ohms")
WaitTime (24,"")
PrintHk(445)
```

```
#set the voltage to 2V
Trig(71, 1, 6553)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 3V
Trig(71, 1, 9830)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 4V
Trig(71, 1, 13106)
WaitTime (24,"")
PrintHk(445)

#back to measure only mode
Trig(71, 1, 0)
WaitForGo ("Set the resistor value to 90 ohms")
WaitTime (24,"")
PrintHk(445)

#set the voltage to 2V
Trig(71, 1, 6553)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 3V
Trig(71, 1, 9830)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 4V
Trig(71, 1, 13106)
WaitTime (24,"")
PrintHk(445)

#back to measure only mode
Trig(71, 1, 0)
WaitForGo ("Set the resistor value to 120 ohms")
WaitTime (24,"")
PrintHk(445)

#set the voltage to 2V
Trig(71, 1, 6553)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 3V
Trig(71, 1, 9830)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 4V
Trig(71, 1, 13106)
WaitTime (24,"")
PrintHk(445)

#back to measure only mode
Trig(71, 1, 0)
WaitForGo ("Set the resistor value to 150 ohms")
WaitTime (24,"")
PrintHk(445)

#set the voltage to 2V
Trig(71, 1, 6553)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 3V
Trig(71, 1, 9830)
WaitTime (24,"")
PrintHk(445)

#set the voltage to 4V
Trig(71, 1, 13106)
WaitTime (24,"")
PrintHk(445)
```

```
#switch off BB1
Trig (69,0,0)

#same test on the calibration source 2
WaitForGo ("For tests with proto: Replace the calibration source 2 by a variable resistor")

#switch on BB2
Trig (72,0,0)
WaitForGo ("Set the resistor value to 30 ohms")
WaitTime (24,"")
PrintHk(447)

#set the voltage to 2V
Trig(75, 1, 6553)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 3V
Trig(75, 1, 9830)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 4V
Trig(75, 1, 13106)
WaitTime (24,"")
PrintHk(447)

#back to measure only mode
Trig(75, 1, 0)
WaitForGo ("Set the resistor value to 60 ohms")
WaitTime (24,"")
PrintHk(447)

#set the voltage to 2V
Trig(75, 1, 6553)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 3V
Trig(75, 1, 9830)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 4V
Trig(75, 1, 13106)
WaitTime (24,"")
PrintHk(447)

#back to measure only mode
Trig(75, 1, 0)
WaitForGo ("Set the resistor value to 90 ohms")
WaitTime (24,"")
PrintHk(447)

#set the voltage to 2V
Trig(75, 1, 6553)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 3V
Trig(75, 1, 9830)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 4V
Trig(75, 1, 13106)
WaitTime (24,"")
PrintHk(447)

#back to measure only mode
Trig(75, 1, 0)
WaitForGo ("Set the resistor value to 120 ohms")
WaitTime (24,"")
PrintHk(447)

#set the voltage to 2V
Trig(75, 1, 6553)
```

```
WaitTime (24,"")
PrintHk(447)

#set the voltage to 3V
Trig(75, 1, 9830)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 4V
Trig(75, 1, 13106)
WaitTime (24,"")
PrintHk(447)

#back to measure only mode
Trig(75, 1, 0)
WaitForGo ("Set the resistor value to 150 ohms")
WaitTime (24,"")
PrintHk(447)

#set the voltage to 2V
Trig(75, 1, 6553)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 3V
Trig(75, 1, 9830)
WaitTime (24,"")
PrintHk(447)

#set the voltage to 4V
Trig(75, 1, 13106)
WaitTime (24,"")
PrintHk(447)

#switch off BB2
Trig (73,0,0)


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_bb_nom.xls")
```

BB.3: Calibration source stability
Script file: a_bb_stab.txt

```
Log("CALIBRATION SOURCE STABILITY MEASURE")

WaitForGo("STABILITY MEASURE: Connect P15 CAL SRC SIM")

#switch on BB1
Trig(68,0,0)

#switch on BB2
Trig(72,0,0)

#set the voltage to 3V
Trig(71, 1, 9830)
Trig(75, 1, 9830)

WaitForGo("click here when the test is finished")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_bb_stab.xls")
```

FW.1: Filter wheel controller
Script file: a_fw.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#Make sure we can execute the script: switch-off FW, stop hk diag if it was running
Trig(39, 0, 0)
Trig(77, 0, 0)
```

```
WaitForGo("If working with proto wheel, make sure the wheel is connected to J13");

#############################################
Log("Test Spectro FW in spectro timing mode")
#############################################

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)

#reset default parameters
Write6(17, 100, 4096, 2000, 2000, 2000, 2000)

# Switch-on the spectro filter wheel
Trig(58, 0, 0)
WaitTime(0.5, "")

# First make sure that the wheel is in position A
# Move it to position A
Trig(64, 1, 0)
WaitForGo("Press OK when the wheel is in position A")

#try invalid parameters
Trig(64, 1, 4)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()
Trig(64, 1, -1)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#start a hk diag at 1KHz with DMC_FWSPEC_POS_A, DMC_FWSPEC_POS_B, DMC_FW_GR_VMOTA,
#DMC_FW_GR_VMOTB, DMC_FW_GR_IMOTA, DMC_FW_GR_IMOTB, DMC_FWGRAT_HALLA, DMC_FWGRAT_HALLB,
# for further analysis by CSL
Write9(13, 555, 559, 556, 560, 564, 567, 256, 257, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 0)

# Move it to position B
Trig(64, 1, 1)
WaitTime(4, "")

# During the move, check status bits in DMC_FW_SPEC_CTRL_ST: you should see bit25=1 and
bit27=1.
# bit 25    DMC_FWSC_MOVING         1 = Currently moving        / 0 = Currenlty not moving
# bit 26    DMC_FWSC_SEARCH_A 1 = Searching position A    / 0 = Not searching position A
# bit 27    DMC_FWSC_SEARCH_B 1 = Searching position B    / 0 = Not searching position B
# bit 28    DMC_FWSC_POS_A          1 = Currently at position A / 0 = Currenlty not at
position A
# bit 29    DMC_FWSC_POS_B          1 = Currently at position B / 0 = Currenlty not at
position B

ChkBit(210, 25, 1)
ChkBit(210, 26, 0)
ChkBit(210, 27, 1)
ChkBit(210, 28, 0)
ChkBit(210, 29, 0)

WaitForGo("Press OK when the wheel is in position B")
WaitTime(2, "")

# At the end of the move, check status bits in DMC_FW_SPEC_CTRL_ST : you should see bit29=1
ChkBit(210, 25, 0)
ChkBit(210, 26, 0)
ChkBit(210, 27, 0)
ChkBit(210, 28, 0)
ChkBit(210, 29, 1)

# Move it to position A in the opposite direction
Trig(64, 1, 2)
WaitTime(4, "")
# During the move, check status bits in DMC_FW_SPEC_CTRL_ST: you should see bit25=1 and
bit26=1.
ChkBit(210, 25, 1)
ChkBit(210, 26, 1)
ChkBit(210, 27, 0)
ChkBit(210, 28, 0)
ChkBit(210, 29, 0)

WaitForGo("Press OK when the wheel is in position A")
WaitTime(2, "")
# At the end of the move, check status bits in DMC_FW_SPEC_CTRL_ST : you should see bit28=1
ChkBit(210, 25, 0)
ChkBit(210, 26, 0)
ChkBit(210, 27, 0)
ChkBit(210, 28, 1)
ChkBit(210, 29, 0)

# Move it to position B in the opposite direction
Trig(64, 1, 3)
WaitForGo("Press OK when the wheel is in position B")

# Move the FW by 1/2 turn
Trig(65, 1, 0x300)
WaitTime(15, "")

# After the command, bit28=1 (pos A)
ChkBit(210, 28, 1)
ChkBit(210, 29, 0)


#enter the simulation mode
Trig(90, 1, 0x04)

# Move it to position B
Trig(64, 1, 1)
WaitTime(3, "")
ChkBit(210, 25, 0)
ChkBit(210, 26, 0)
ChkBit(210, 27, 0)
ChkBit(210, 28, 0)
ChkBit(210, 29, 1)

# Move it to position A
Trig(64, 1, 0)
WaitTime(3, "")
ChkBit(210, 25, 0)
ChkBit(210, 26, 0)
ChkBit(210, 27, 0)
ChkBit(210, 28, 1)
ChkBit(210, 29, 0)

#exit simulation mode
Trig(90, 1, 0x00)

# now, let's test the thresholds
#*******************************

# set high threshold (bigger than the max value of the sensor) for the control and low
threshold for the status
Write6(17, 100, 4096, 2000, 15000, 2000, 2000)
#try to move it to B (it should not work since the control threshold will never be reached)
Trig(64, 1, 1)
ManualCheck("Press OK if the wheel did not stop on position B")

# set low threshold for the control and high threshold for the status
Write6(17, 100, 4096, 2000, 2000, 2000, 15000)
#try to move it to B (it should not work since the control threshold will never be reached)
Trig(64, 1, 1)
ManualCheck("Press OK if the wheel is on position B")
ChkBit(210, 28, 0) //in the HK, we can not see that the position is reached
ChkBit(210, 29, 0)

# set high threshold (bigger than the max value of the sensor) for the control and low
threshold for the status
Write6(17, 100, 4096, 15000, 2000, 2000, 2000)
#try to move it to A (it should not work since the control threshold will never be reached)
Trig(64, 1, 0)
ManualCheck("Press OK if the wheel did not stop on position A")

# set low threshold for the control and high threshold for the status
Write6(17, 100, 4096, 2000, 2000, 15000, 2000)
#try to move it to A (it should not work since the control threshold will never be reached)
Trig(64, 1, 0)
ManualCheck("Press OK if the wheel is on position A")
```

```
ChkBit(210, 28, 0) //in the HK, we can not see that the position is reached
ChkBit(210, 29, 0)

#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_fw_spec.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_fw_spec.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

#switch-off FW
Trig(39, 0, 0)

#reset default parameters
Write6(17, 100, 4096, 2000, 2000, 2000, 2000)

WaitForGo("If working with proto wheel, make sure the wheel is connected to J14");

#############################################
Log("Test Photo FW in photo timing mode")
#############################################

#go to photo mode (timing only)
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#reset default parameters
Write6(18, 100, 4096, 2000, 2000, 2000, 2000)

# Switch-on FW Photo
Trig(59, 0, 0)
WaitTime(0.5, "")

# First make sure that the wheel is in position A
# Move it to position A
Trig(66, 1, 0)
WaitForGo("Press OK when the wheel is in position A")

#try invalid parameters
Trig(66, 1, 4)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()
Trig(66, 1, -1)
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

#start a hk diag at 1KHz with DMC_FWPHOT_POS_A, DMC_FWPHOT_POS_B, DMC_FW_GR_VMOTA,
#DMC_FW_GR_VMOTB, DMC_FW_GR_IMOTA, DMC_FW_GR_IMOTB, DMC_FWGRAT_HALLA, DMC_FWGRAT_HALLB,
# for further analysis by CSL
Write9(13, 563, 569, 556, 560, 564, 567, 256, 257, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 0)

# Move it to position B
Trig(66, 1, 1)
WaitTime(4, "")

# During the move, check status bits in DMC_FW_SPEC_CTRL_ST: you should see bit25=1 and
bit27=1.
ChkBit(211, 25, 1)
ChkBit(211, 26, 0)
ChkBit(211, 27, 1)
ChkBit(211, 28, 0)
ChkBit(211, 29, 0)

WaitForGo("Press OK when the wheel is in position B")
WaitTime(2, "")

# At the end of the move, check status bits in DMC_FW_SPEC_CTRL_ST : you should see bit29=1
ChkBit(211, 25, 0)
ChkBit(211, 26, 0)
ChkBit(211, 27, 0)
ChkBit(211, 28, 0)
ChkBit(211, 29, 1)

# Move it to position A in the reverse direction
Trig(66, 1, 2)
WaitTime(4, "")

# During the move, check status bits in DMC_FW_SPEC_CTRL_ST: you should see bit25=1 and
bit26=1.
ChkBit(211, 25, 1)
ChkBit(211, 26, 1)
ChkBit(211, 27, 0)
ChkBit(211, 28, 0)
ChkBit(211, 29, 0)

WaitForGo("Press OK when the wheel is in position A")
WaitTime(2, "")

# At the end of the move, check status bits in DMC_FW_SPEC_CTRL_ST : you should see bit28=1
ChkBit(211, 25, 0)
ChkBit(211, 26, 0)
ChkBit(211, 27, 0)
ChkBit(211, 28, 1)
ChkBit(211, 29, 0)

# Move it to position B in the reverse direction
Trig(66, 1, 3)
WaitForGo("Press OK when the wheel is in position B")

# Move the FW by 1/2 turn
Trig(67, 1, 0x300)
WaitTime(15, "")

# After the command, bit28=1
ChkBit(211, 28, 1)
ChkBit(211, 29, 0)

#enter the simulation mode
Trig(90, 1, 0x08)

# Move it to position B
Trig(66, 1, 1)
WaitTime(3, "")
ChkBit(211, 25, 0)
ChkBit(211, 26, 0)
ChkBit(211, 27, 0)
ChkBit(211, 28, 0)
ChkBit(211, 29, 1)

# Move it to position A
Trig(66, 1, 0)
WaitTime(3, "")
ChkBit(211, 25, 0)
ChkBit(211, 26, 0)
ChkBit(211, 27, 0)
ChkBit(211, 28, 1)
ChkBit(211, 29, 0)

#exit simulation mode
Trig(90, 1, 0x00)


# now, let's test the thresholds
#********************************

# set high threshold (bigger than the max value of the sensor) for the control and low
threshold for the status
Write6(18, 100, 4096, 2000, 15000, 2000, 2000)
#try to move it to B (it should not work since the control threshold will never be reached)
Trig(66, 1, 1)
ManualCheck("Press OK if the wheel did not stop on position B")

# set low threshold for the control and high threshold for the status
Write6(18, 100, 4096, 2000, 2000, 2000, 15000)
#try to move it to B (it should not work since the control threshold will never be reached)
Trig(66, 1, 1)
ManualCheck("Press OK if the wheel is on position B")
```

```
ChkBit(211, 28, 0) //in the HK, we can not see that the position is reached
ChkBit(211, 29, 0)

# set high threshold (bigger than the max value of the sensor) for the control and low
threshold for the status
Write6(18, 100, 4096, 15000, 2000, 2000, 2000)
#try to move it to A (it should not work since the control threshold will never be reached)
Trig(66, 1, 0)
ManualCheck("Press OK if the wheel did not stop on position A")

# set low threshold for the control and high threshold for the status
Write6(18, 100, 4096, 2000, 2000, 15000, 2000)
#try to move it to A (it should not work since the control threshold will never be reached)
Trig(66, 1, 0)
ManualCheck("Press OK if the wheel is on position A")
ChkBit(211, 28, 0) //in the HK, we can not see that the position is reached
ChkBit(211, 29, 0)


#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_fw_phot.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_fw_phot.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

# Try to move FW spectro
Trig(64, 1, 1)
WaitTime(3, "")
ManualCheck("Did DPU received a NACK ?")
DmcIsAlive()

# switch off the FW wheels + grating
Trig(39, 0, 0)
WaitTime(3, "")

# DMC_FW_SPEC_CTRL, bit20 should be 0 (powered off).
ChkBit(211, 20, 0)

# DMC_FW_PHOT_CTRL, bit20 should be 0.
ChkBit(210, 20, 0)

#reset default parameters
Write6(18, 100, 4096, 2000, 2000, 2000, 2000)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_fw_nom.xls")
ChkReport()
```

TS.1: Calibration of temperature sensors
Script file: cal_ts.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Connect the Temperature sensor simulator to the DMC ")
WaitForGo("Connect the 100 Ohms connector to the simulator")
Trig(12, 0, 0)
WaitTime(5,"")
#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)

#start a hk diag at 20Hz
Write12(13, 619, 620, 621, 575, 576, 622, 623, 624, 625, 571, 572, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 50)

Trig(95, 0, 0)
WaitTime(90, "")

ChkGT(295,97)
ChkLT(295,103)
```

```
ChkGT(296,97)
ChkLT(296,103)
ChkGT(329,97)
ChkLT(329,103)
ChkGT(330,97)
ChkLT(330,103)

ChkGT(405,97)
ChkLT(405,103)
ChkGT(406,97)
ChkLT(406,103)
ChkGT(407,97)
ChkLT(407,103)
ChkGT(408,97)
ChkLT(408,103)
ChkGT(426,97)
ChkLT(426,103)
ChkGT(427,97)
ChkLT(427,103)
ChkGT(429,97)
ChkLT(429,103)

WaitForGo("Connect the 500 Ohms connector to the simulator")
WaitTime(60, "")

ChkGT(295,487)
ChkLT(295,513)
ChkGT(296,487)
ChkLT(296,513)
ChkGT(329,487)
ChkLT(329,513)
ChkGT(330,487)
ChkLT(330,513)

ChkGT(405,487)
ChkLT(405,513)
ChkGT(406,487)
ChkLT(406,513)
ChkGT(407,487)
ChkLT(407,513)
ChkGT(408,487)
ChkLT(408,513)
ChkGT(426,487)
ChkLT(426,513)
ChkGT(427,487)
ChkLT(427,513)
ChkGT(429,487)
ChkLT(429,513)

WaitForGo("Connect the 2K/5K Ohms connector to the simulator")
WaitTime(60, "")

ChkGT(295,4870)
ChkLT(295,5130)
ChkGT(296,4870)
ChkLT(296,5130)
ChkGT(329,4870)
ChkLT(329,5130)
ChkGT(330,4870)
ChkLT(330,5130)
ChkGT(405,1950)
ChkLT(405,2050)
ChkGT(406,1950)
ChkLT(406,2050)
ChkGT(407,1950)
ChkLT(407,2050)
ChkGT(408,1950)
ChkLT(408,2050)
ChkGT(426,1950)
ChkLT(426,2050)
ChkGT(427,1950)
ChkLT(427,2050)
ChkGT(429,1950)
ChkLT(429,2050)

WaitForGo("Connect the 5K/15K Ohms connector to the simulator")
WaitTime(60, "")
```

```
ChkGT(295,14625)
ChkLT(295,15375)
ChkGT(296,14625)
ChkLT(296,15375)
ChkGT(329,14625)
ChkLT(329,15375)
ChkGT(330,14625)
ChkLT(330,15375)

ChkGT(405,4870)
ChkLT(405,5130)
ChkGT(406,4870)
ChkLT(406,5130)
ChkGT(407,4870)
ChkLT(407,5130)
ChkGT(408,4870)
ChkLT(408,5130)
ChkGT(426,4870)
ChkLT(426,5130)
ChkGT(427,4870)
ChkLT(427,5130)
ChkGT(429,4870)
ChkLT(429,5130)

#go to photo mode (timing only)
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)

WaitTime(60, "")

ChkGT(405,4870)
ChkLT(405,5130)
ChkGT(406,4870)
ChkLT(406,5130)
ChkGT(407,4870)
ChkLT(407,5130)
ChkGT(408,4870)
ChkLT(408,5130)
ChkGT(426,4870)
ChkLT(426,5130)
ChkGT(427,4870)
ChkLT(427,5130)
ChkGT(429,4870)
ChkLT(429,5130)


#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_Temp_Sens_all.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\cal_ts.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")


Trig(13, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_ts_nom.xls")
ChkReport()
```

TS.2: Calibration of temperature sensors (red DEC + redundant DMC)
Script file: cal_ts_red.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Connect the Temperature sensor simulator to the DMC ")
WaitForGo("Connect the 100 Ohms connector to the simulator")
Trig(19, 0, 0)
WaitTime(5,"")
#go to spectro mode (timing only)
```

```
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)

#start a hk diag at 20Hz
Write12(13, 619, 620, 621, 575, 576, 622, 623, 624, 625, 571, 572, 0xFFFF)
#delete old hk diag file
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Trig(76, 1, 50)

Trig(95, 0, 0)
WaitTime(90, "")

ChkGT(363,97)
ChkLT(363,103)
ChkGT(364,97)
ChkLT(364,103)
ChkGT(397,97)
ChkLT(397,103)
ChkGT(398,97)
ChkLT(398,103)
ChkGT(405,97)
ChkLT(405,103)
ChkGT(406,97)
ChkLT(406,103)
ChkGT(407,97)
ChkLT(407,103)
ChkGT(408,97)
ChkLT(408,103)
ChkGT(426,97)
ChkLT(426,103)
ChkGT(427,97)
ChkLT(427,103)
ChkGT(429,97)
ChkLT(429,103)

WaitForGo("Connect the 500 Ohms connector to the simulator")
WaitTime(60, "")

ChkGT(363,487)
ChkLT(363,513)
ChkGT(364,487)
ChkLT(364,513)
ChkGT(397,487)
ChkLT(397,513)
ChkGT(398,487)
ChkLT(398,513)
ChkGT(405,487)
ChkLT(405,513)
ChkGT(406,487)
ChkLT(406,513)
ChkGT(407,487)
ChkLT(407,513)
ChkGT(408,487)
ChkLT(408,513)
ChkGT(426,487)
ChkLT(426,513)
ChkGT(427,487)
ChkLT(427,513)
ChkGT(429,487)
ChkLT(429,513)

WaitForGo("Connect the 2K/5K Ohms connector to the simulator")
WaitTime(60, "")

ChkGT(363,4870)
ChkLT(363,5130)
ChkGT(364,4870)
ChkLT(364,5130)
ChkGT(397,4870)
ChkLT(397,5130)
ChkGT(398,4870)
ChkLT(398,5130)
ChkGT(405,1950)
ChkLT(405,2050)
ChkGT(406,1950)
ChkLT(406,2050)
ChkGT(407,1950)
ChkLT(407,2050)
```

```
ChkGT(408,1950)
ChkLT(408,2050)
ChkGT(426,1950)
ChkLT(426,2050)
ChkGT(427,1950)
ChkLT(427,2050)
ChkGT(429,1950)
ChkLT(429,2050)

WaitForGo("Connect the 5K/15K Ohms connector to the simulator")
WaitTime(60, "")

ChkGT(363,14625)
ChkLT(363,15375)
ChkGT(364,14625)
ChkLT(364,15375)
ChkGT(397,14625)
ChkLT(397,15375)
ChkGT(398,14625)
ChkLT(398,15375)
ChkGT(405,4870)
ChkLT(405,5130)
ChkGT(406,4870)
ChkLT(406,5130)
ChkGT(407,4870)
ChkLT(407,5130)
ChkGT(408,4870)
ChkLT(408,5130)
ChkGT(426,4870)
ChkLT(426,5130)
ChkGT(427,4870)
ChkLT(427,5130)
ChkGT(429,4870)
ChkLT(429,5130)

#go to photo mode (timing only)
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)

WaitTime(60, "")

ChkGT(363,14625)
ChkLT(363,15375)
ChkGT(364,14625)
ChkLT(364,15375)
ChkGT(397,14625)
ChkLT(397,15375)
ChkGT(398,14625)
ChkLT(398,15375)
ChkGT(405,4870)
ChkLT(405,5130)
ChkGT(406,4870)
ChkLT(406,5130)
ChkGT(407,4870)
ChkLT(407,5130)
ChkGT(408,4870)
ChkLT(408,5130)
ChkGT(426,4870)
ChkLT(426,5130)
ChkGT(427,4870)
ChkLT(427,5130)
ChkGT(429,4870)
ChkLT(429,5130)


#stop hk diag and save it for further analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_Temp_Sens_all.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\cal_ts_red.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

Trig(20, 0, 0)
```

```
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_ts_red.xls")
ChkReport()
```

PHD.1.a: Photoconducting detectors control (red DEC)
Script file: a_heat_flash_red.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#*****************************************
#HEATER
#*****************************************

WaitForGo("Connect P78 connector to J178 to simulate heater and flasher.  Connect a jumper
between 8-15 and an amperemeter between 7-14")
Trig(19, 0, 0)
WaitTime(8, "")

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 1)

#enable the heater
Trig(79, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, -20)
ChkLT(349, 20)
ManualCheck("Check that the current is close to zero")

#check voltage in HK
ChkGT(350, -50)
ChkLT(350, 50)

#set a 10mA current in the heater
Trig(34, 1, 0x800)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, 12700)
ChkLT(349, 13500)
ManualCheck("Check that the current is 10mA +- 3%")

#set a 20mA current in the heater
Trig(34, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, 25400)
ChkLT(349, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

WaitForGo("Connect the 1300 ohms resistor between 7 and the amperemeter")
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, 25400)
ChkLT(349, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

#check voltage in HK
ChkGT(350, 22100)
ChkLT(350, 23300)

#switch-off heater
Trig(80, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, -20)
ChkLT(349, 20)
```

```
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(350, -50)
ChkLT(350, 50)

#set current to 20mA while heater is off
Trig(34, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(349, -20)
ChkLT(349, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(350, -50)
ChkLT(350, 50)

#switch-off red DEC
Trig(20, 0, 0)

#****************************************
#FLASHER
#****************************************

WaitForGo("Connect P78 connector to J178 to simulate heater and flasher.  Connect a jumper
between 7-14 and an amperemeter between 8-15")
Trig(19, 0, 0)
WaitTime(8, "")

#enable the flasher
Trig(81, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(383, -20)
ChkLT(383, 20)
ManualCheck("Check that the current is close to zero")

#check voltage in HK
ChkGT(384, -50)
ChkLT(384, 50)

#set a 10mA current in the flasher
Trig(35, 1, 0x800)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(383, 12700)
ChkLT(383, 13500)
ManualCheck("Check that the current is 10mA +- 3%")

#set a 20mA current in the flasher
Trig(35, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(383, 25400)
ChkLT(383, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

WaitForGo("Connect the 1300 ohms resistor between 8 and the amperemeter")
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(383, 25400)
ChkLT(383, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

#check voltage in HK
ChkGT(384, 22100)
ChkLT(384, 23300)

#switch-off flasher
Trig(82, 0, 0)
WaitTime(2.5, "")
```

```
#check current in HK and on the amperemeter
ChkGT(383, -20)
ChkLT(383, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(384, -50)
ChkLT(384, 50)

#set current to 20mA while flasher is off
Trig(35, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(383, -20)
ChkLT(383, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(384, -50)
ChkLT(384, 50)

Trig(20, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_heat_flash_red.xls")
ChkReport()
```

PHD.1.b: Photoconducting detectors control (blue DEC)
Script file: a_heat_flash.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#****************************************
#HEATER
#****************************************

WaitForGo("Connect P78 connector to J78 to simulate heater and flasher.  Connect a jumper
between 8-15 and an amperemeter between 7-14")
Trig(12, 0, 0)
WaitTime(8, "")

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 1)

#enable the heater
Trig(60, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(281, -20)
ChkLT(281, 20)
ManualCheck("Check that the current is close to zero")

#check voltage in HK
ChkGT(282, -50)
ChkLT(282, 50)

#set a 10mA current in the heater
Trig(17, 1, 0x800)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(281, 12700)
ChkLT(281, 13500)
ManualCheck("Check that the current is 10mA +- 3%")

#set a 20mA current in the heater
Trig(17, 1, 0xFFF)
WaitTime(2.5, "")
```

```
#check current in HK and on the amperemeter
ChkGT(281, 25400)
ChkLT(281, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

WaitForGo("Connect the 1300 ohms resistor between 7 and the amperemeter")
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(281, 25400)
ChkLT(281, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

#check voltage in HK
ChkGT(282, 22100)
ChkLT(282, 23300)

#switch-off heater
Trig(61, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(281, -20)
ChkLT(281, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(282, -50)
ChkLT(282, 50)

#set current to 20mA while heater is off
Trig(17, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(281, -20)
ChkLT(281, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(282, -50)
ChkLT(282, 50)

#switch-off blue DEC
Trig(13, 0, 0)

#****************************************
#FLASHER
#****************************************

WaitForGo("Connect P78 connector to J78 to simulate heater and flasher.  Connect a jumper
between 7-14 and an amperemeter between 8-15")
Trig(12, 0, 0)
WaitTime(8, "")

#enable the flasher
Trig(62, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, -20)
ChkLT(315, 20)
ManualCheck("Check that the current is close to zero")

#check voltage in HK
ChkGT(316, -50)
ChkLT(316, 50)

#set a 10mA current in the flasher
Trig(18, 1, 0x800)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, 12700)
ChkLT(315, 13500)
ManualCheck("Check that the current is 10mA +- 3%")
```

```
#set a 20mA current in the flasher
Trig(18, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, 25400)
ChkLT(315, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

WaitForGo("Connect the 1300 ohms resistor between 8 and the amperemeter")
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, 25400)
ChkLT(315, 27000)
ManualCheck("Check that the current is 20mA +- 3%")

#check voltage in HK
ChkGT(316, 22100)
ChkLT(316, 23300)

#switch-off flasher
Trig(63, 0, 0)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, -20)
ChkLT(315, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(316, -50)
ChkLT(316, 50)

#set current to 20mA while flasher is off
Trig(18, 1, 0xFFF)
WaitTime(2.5, "")

#check current in HK and on the amperemeter
ChkGT(315, -20)
ChkLT(315, 20)
ManualCheck("Check that the current is close to 0mA")

#check voltage in HK
ChkGT(316, -50)
ChkLT(316, 50)

Trig(13, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_heat_flash_nom.xls")
ChkReport()
```

PHA.1: Photoconducting arrays and SPU interface: sending data to blue SPU
Script file: a_dec2spu.txt

```
Before starting the test, connect the SPU Sim cable to J02
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Start the Link Receiver to replace the Blue SPU, configure it to receive packets
and throw them [option 5]");

# switch on blue DEC
Trig(12, 0, 0)
WaitTime(5, "")

# Start the link with blue SPU
Trig(87,1,1 )

#set the SPU transmission mode
Write2(28, 0xAAAA, 0xBBBB)

# Forward the science data from Blue DEC to SPU
```

```
Write1(23, 0)
WaitTime(8, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In hk.xls, check that DMC_DECB_REC_PAC and DMC_BLUE_ENC_PAC are incrementing
by 512 between 2 HK packets")

#stop forwarding science data from Blue DEC to SPU
Write1(23, 4)

WaitForGo("Stop the Link Receiver and restart it, configure it to receive 10 packets in
circular buffering mode [option 3]");

#set Blue DEC in simulator mode
Write3 (27,0x20,0x8,0x1CC)
WaitTime(5, "")
Trig (16,0,0)
WaitTime(2, "")

# Start the link with blue SPU
Trig(87,1,1 )

# Forward the science data from Blue DEC to SPU during 2 seconds
Write1(23, 0)
WaitTime(2, "")
Write1(23, 4)

ManualCheck("On the SPU Sim, open one of the saved file with an HEX editor and check that
the packet is compliant with the SPU-DMC ICD");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_dec2spu_nom.xls")
ChkReport()
```

PHA.2: Photoconducting arrays and SPU interface: sending data to red SPU
Script file: a_dec2spu2.txt

```
Before starting the test, connect the SPU Sim cable to J03
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Start the Link Receiver to replace the Red SPU, configure it to receive 10
packets in circular buffering mode [option 3]");

# switch on blue DEC
Trig(12, 0, 0)
WaitTime(6, "")

#set the SPU transmission mode
Write2(28, 0xAAAA, 0xBBBB)

#set Blue DEC in simulator mode
Write3 (27,0x20,0x8,0x1CC)
WaitTime(5, "")
Trig (16,0,0)
WaitTime(2, "")
# Start the link with red SPU
Trig(86,1,1 )

# Transfer Blue data to SPU red output
Write1(30, 3)

# Forward the science data from Blue DEC to SPU
Write1(23, 0)

ManualCheck("On the SPU Sim, open one of the saved file with an HEX editor and check that
the packet is compliant with the SPU-DMC ICD, you should especially check the APID");


# Stop forwarding the science data from Blue DEC to SPU
Write1(23, 4)

# Switch off blue DEC
```

```
Trig(13, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_dec2spu2_nom.xls")
ChkReport()
```

PHA.3: Photoconducting arrays and SPU interface: transfer data from detector simulator
Script file: a_dec2spu3.txt

```
Before starting the test, connect the SPU Sim cable to J03
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Start the Link Receiver to replace the Red SPU, configure it to save packets on
disk [circular numbering with 100 files [option 3]]");

#start link with red SPU
Trig(86,1,1 )

#start detector simulator to simulate red DEC at 100Hz
Trig(31, 1, 0x2000000A )

#forward science data from red DEC to SPU
Write1(24, 0)
WaitTime(10, "")

#stop the detector simulator
Trig(32,0,0 )

ManualCheck("Open one of the saved file with an HEX editor and check that the packet is
compliant with the SPU-DMC ICD");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In hk.xls, DMC_DECR_REC_PAC shall increment by 200 between 2 hk packets and
stop incrementing after the stop command has been received");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_dec2spu3_nom.xls")
ChkReport()
```

BOA.1: Bolometers arrays and SPU interface: transfer data to blue SPU
Script file: a_bol2spu.txt

```
Before starting the test, connect the Blue SPU Sim cable to J02, and restart DMC
```

```
ChkClearReport ()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#set the timing FPGA in photometry mode
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#set the SPU transmission mode
Write2(28, 0xAAAA, 0xBBBB)

WaitForGo("Start the Link Receiver to replace the BLUE SPU, configure it to receive 256
packets and save them to file [option 4]")

# Start the link with blue SPU
Trig(87,1,1 )

# Reset SMCS chip 2 and connect to BOLC
Trig(89,0,0 )
WaitTime(2, "")
```

```
# Configure the BOLC to send science data at 40Hz
Trig(33, 1, 0x09020002)
WaitTime(0.5, "")
Trig(33, 1, 0x0B020000)
WaitTime(10, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("Launch 'View hk.bat', In 'Hk.xls', DMC_BOL_REC_PAC shall increment by 480
between 2 hk packets.")

# Forward the data from BOLC to SPU
Write1(22, 0)
WaitTime(5, "")

ManualCheck("Open one of the saved file with an HEX editor and check that the packet is
compliant with the SPU-DMC ICD")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_bolc2spu_nom.xls")

ChkReport()
```

BOA.2: Bolometers arrays and SPU interface: transfer data from detector simulator to red SPU
Script file: a_bol2spu2.txt

```
Before starting the test, connect the Red SPU Sim cable to J03, restart DMC, switch-off
BOLC
```

```
ChkClearReport ()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Make sure BOLC is switched-off")
WaitForGo("Start the Link Receiver to replace the RED SPU, configure it to receive 256
packets and save them to file [option 4]")

#set the SPU transmission mode
Write2(28, 0xAAAA, 0xBBBB)

# start the link with red SPU
Trig(86,1,1 )

# start simulating BOLC data at 33Hz
Trig(31,1,0x40000020)

# Forward the data from BOLC to SPU
Write1(22, 0)
WaitTime(5, "")

ManualCheck("Open one of the saved file with an HEX editor and check that the packet is
compliant with the SPU-DMC ICD")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_bolc2spu2_nom.xls")

ChkReport()
```

TIM.1: Timing: OBT counter
Script file: a_mim1fpga.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")

#####################################
#Test with nominal OBT frequency
#####################################

WaitForGo("Make sure OBT frequency is 131072Hz");

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 1)

#set the hk diag list to OBT_COUNT and ISR_COUNT
Write3(13, 453, 242, 0xffff)

#switch-on DECs
Trig(12, 0, 0)
Trig(19, 0, 0)
WaitTime(5, "")

#connect to BOLC
Trig(89, 0, 0)
WaitTime(1, "")

#synchronize on red DEC
Trig(10, 1, 2)
WaitTime(2, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(1, "")
Trig(77, 0, 0)
```

```
System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 512
[=131072/256]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

#change blue DEC readout frequency to 128Hz
Write1(27, 64)
Trig(16, 0, 0)
WaitTime(2, "")

#synchronize on blue DEC
Trig(10, 1, 1)
WaitTime(2, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(1, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 1024
[=131072/128]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

#go to photo mode (timing only).  Note, for this test, to have the best accuracy, we use a
phase_inc that has been adapted to the BOLC Sim frequency.
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#configure BOLC to send readouts at 40Hz
Trig(33, 1, 0x09020002)
WaitTime(0.5, "")
Trig(33, 1, 0x0B020000)
WaitTime(2, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(1, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3277 or
3276 [=131072/40]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

#configure BOLC to send readouts at 2Hz
Trig(33, 1, 0x0B020260)
WaitTime(2, "")

#start HK diag for 5 sec
Trig(76, 1, 20)
WaitTime(5, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 65536
[=131072/2]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, The DMC_ISR_COUNT shall increment by 16640")
```

```
#####################################
#Test PLL with nominal OBT frequency
#####################################

#configure BOLC to send readouts at 40Hz
Trig(33, 1, 0x0B020000)
WaitTime(3, "")

#check PLL residue
ChkLT(264, 16641)
ChkGT(264, 16638)
PrintHk(263)

#configure BOLC to send readouts at 20Hz
Trig(33, 1, 0x0B020020)
WaitTime(3, "")

#check PLL residue
ChkLT(264, 33281)
ChkGT(264, 33278)
PrintHk(263)

#configure BOLC to send readouts at 10Hz
Trig(33, 1, 0x0B020060)
WaitTime(3, "")

#check PLL residue (it should be 66560) but since this value is coded on 16bits, it should
be 1024
ChkLT(264, 1026)
ChkGT(264, 1022)
PrintHk(263)

#configure BOLC to send readouts at 2Hz
Trig(33, 1, 0x0B020260)
WaitTime(3, "")

#check PLL residue (it should be 332800) but since this value is coded on 16bits, it should
be 5120
ChkLT(264, 5122)
ChkGT(264, 5118)
PrintHk(263)


#####################################
#Test with modified OBT frequency
#####################################

WaitForGo("Make sure OBT frequency is 130000Hz");
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 2)

#set the hk diag list to OBT_COUNT and ISR_COUNT
Write3(13, 453, 242, 0xffff)

#synchronize on blue DEC
Trig(10, 1, 1)

#change readout frequency to 256Hz
Write1(27, 32)
WaitTime(0.5, "")
Trig(16, 0, 0)
WaitTime(1, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(2, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall still increment by 512
[the readout frequency is linked to OBT frequency]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
```

```
WaitForGo("close hkDiag.xls")

#change readout frequency to 128Hz
Write1(27, 64)
Trig(16, 0, 0)

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(2, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 1024
[=131072/128]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, The DMC_ISR_COUNT shall increment by 16250
[=2*8192*130000/131072]")

#don't lock on OBT anymore
Write6(29, 0, 0, 0, 26, 0x095217cb, 0x3)
Trig(11, 0, 0)
WaitTime(1, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(2, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 1015 or
1016 [=130000/128]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, The DMC_ISR_COUNT shall increment again by 16384")

#go to photo mode (timing only)
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#configure BOLC to send readouts at 40Hz
Trig(33, 1, 0x09020002)
Trig(33, 1, 0x0B020000)
WaitTime(1, "")

#start HK diag for 1 sec
Trig(76, 1, 0)
WaitTime(1, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3250
[=130000/40]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

#configure BOLC to send readouts at 2Hz
Trig(33, 1, 0x0B020260)
WaitTime(1, "")

#start HK diag for 2 sec
Trig(76, 1, 20)
WaitTime(2, "")
```

```
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_OBT.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 65000
[=130000/2]")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

#switch-off blue DEC
Trig(13, 0, 0)

WaitForGo("Make sure OBT frequency is 131072Hz");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_mim1fpga_nom.xls")
ChkReport()
```

TIM.1: Timing: OBT counter
Script file: a_synchro.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")

#####################################
#Test with nominal OBT frequency
#####################################

WaitForGo("Make sure OBT frequency is 131072Hz and BOLC is switched ON");

############# INTERNAL SYNC AT 40Hz
#go to photo mode (timing only) and set internal sync at 40Hz.  Note, for this test, to
have the best accuracy, we use a phase_inc that has been adapted to the BOLC Sim frequency.
Write6(29, 0, 0, 0, 13, 0x09775aa7, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 13, 0x09775aa7, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#set the hk diag list to DMC_OBT_COUNT DMC_SYNC_COUNT DMC_CUSTOM_HK1 DMC_ISR_COUNT
DMC_PLL_RES_HI DMC_PLL_RES_LO
Write7(13, 453, 240, 433, 242, 264, 263, 0xffff)

#connect to BOLC
Trig(89, 0, 0)
WaitTime(1, "")

#synchronize on the internal source
Trig(10, 1, 0xE00)
WaitTime(2, "")

#start HK diag for 5 sec
Trig(76, 1, 0)
WaitTime(5, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_SYNCHRO.bat")
WaitTime(1, "")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3277 or
3276 [=131072/40], DMC_SYNC_COUNT and CUSTOM_HK1 shall increment at the same time [every 25
samples [=1000/40]]")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_synchro_internal40.xls")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")


############# EXTERNAL BOLC SYNC

#go to photo mode (timing only).  Note, for this test, to have the best accuracy, we use a
phase_inc that has been adapted to the BOLC Sim frequency.
```

```
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 0, 0, 26, 0x09775aa7, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#configure BOLC to send readouts at 40Hz
Trig(33, 1, 0x09020002)
WaitTime(0.5, "")
Trig(33, 1, 0x0B020000)
WaitTime(2, "")

#synchronize on BOLC
Trig(10, 1, 4)
WaitTime(2, "")

#start HK diag for 5 sec
Trig(76, 1, 0)
WaitTime(5, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_SYNCHRO.bat")
WaitTime(1, "")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 3277 or
3276 [=131072/40], DMC_SYNC_COUNT and CUSTOM_HK1 shall increment at the same time [every 25
samples [=1000/40]]")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_synchro_bolc.xls")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")


############# EXTERNAL BLUE DEC

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 1)

#set the hk diag list to DMC_OBT_COUNT DMC_SYNC_COUNT DMC_CUSTOM_HK1 DMC_ISR_COUNT
DMC_PLL_RES_HI DMC_PLL_RES_LO
Write7(13, 453, 240, 433, 242, 264, 263, 0xffff)

#switch-on DECs
Trig(12, 0, 0)
Trig(19, 0, 0)
WaitTime(5, "")

#synchronize on blue DEC
Trig(10, 1, 1)
WaitTime(2, "")

#start HK diag for 5 sec
Trig(76, 1, 0)
WaitTime(5, "")
Trig(77, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_SYNCHRO.bat")
WaitTime(1, "")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 512
[=131072/256], DMC_SYNC_COUNT and CUSTOM_HK1 shall increment at the same time [every 3/4
samples [=1000/256]]")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_synchro_blue_dec.xls")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")

############# EXTERNAL RED DEC

#synchronize on red DEC
Trig(10, 1, 2)
WaitTime(2, "")
```

```
#start HK diag for 5 sec
Trig(76, 1, 0)
WaitTime(5, "")
Trig(77, 0, 0)

#switch-off DECs
Trig(13, 0, 0)
Trig(20, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK_diag_SYNCHRO.bat")
WaitTime(1, "")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("In HkDiag.xls, each time OBT_COUNT increments, it shall increment by 512
[=131072/256], DMC_SYNC_COUNT and CUSTOM_HK1 shall increment at the same time [every 3/4
samples [=1000/256]]")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_synchro_red_dec.xls")
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
WaitForGo("close hkDiag.xls")


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_synchro_nom.xls")
ChkReport()
```

TIM.2: Timing: Shifted synchro to trigger mechanisms move
Script file: a_mec_sync.txt

```
Log("* Mech synchro test")

# upload the sequence simpleChop.seq
# DMC_LOOP , 100
#   DMC_MOVE_CHOP_ABS , 13434
#   DMC_WAIT , 3
#   DMC_MOVE_CHOP_ABS , -13434
#   DMC_WAIT , 3
# DMC_END_LOOP , 0
# DMC_MOVE_CHOP_ABS , 0
# DMC_END_SEQUENCE , 0
UploadSeq("simplechop")

#connect to BOLC
Trig(89, 0, 0)

#go to photo mode (timing only) with a phase shift of 80
Write6(29, 0, 80, 0, 26, 0x0977602a, 0x33)
Trig(11, 0, 0)
WaitTime(1, "")
Trig(10, 1, 4)
WaitTime(1.5, "")
Write6(29, 0, 80, 0, 26, 0x0977602a, 0x23)
Trig(11, 0, 0)
WaitTime(1, "")

#configure BOLC to send readouts at 40Hz
Trig(33, 1, 0x09020002)
WaitTime(0.5, "")
Trig(33, 1, 0x0B020000)
WaitTime(2, "")

#set the chopper parameters for DM Tamb
Write21(16, 535080, 33261737, 304, 1114, 293, 0x3FFFFFFF, 0x7fff, 29000, 0x7FFFFFFF, -610,
140000, 0, 0, 101000, 172000, 101000, 1839000, 900000, 137, 652000, 8000)

#switch on the chopper
Trig(49, 0, 0)
WaitTime(0.5, "")

# change the hk diag list
Write5(13, 244, 245, 240 , 433, 0xFFFF)

WaitForGo("Launch HkDiagRTViewer")

# start diag hk at 1Khz
Trig(76, 1, 0)
```

```
# enable the chopper controller
Trig(51, 0, 0)

# start the sequence
Trig(5, 0, 0)
WaitForGo("Wait sequence is completed")
Trig(6, 0, 0)

# stop hk diag
Trig(77, 0, 0)
WaitTime(0.5, "")

System("D:\prj\PACS\TestPlanTools\View_HK_diag_mec_sync.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")
ManualCheck("Chopper shall start moving only 10 samples after the DMC_SYNC_COUNT has
changed ?")
WaitForGo("close hkdiag.xls");
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_mec_sync.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")


#disable chopper controller
Trig(52, 0, 0)
WaitTime(0.5, "")

#switch off the chopper
Trig(50, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_mec_sync_nom.xls")
ChkReport()
```

INT.1: Interface
Script file: a_int.txt

```
ChkClearReport ()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Start a Blue SPU Simulator and configure it to receive packets and throw them
[option 5]")

#check that the 1355 is not connected
ChkBit(202, 19, 1)

#Start connection with Blue SPU
Trig(87, 1, 1)
WaitTime(2.5, "")

#check that the 1355 is connected
ChkBit(202, 19, 0)

WaitForGo("disconnect the cable between Blue SPU and DMC [J02 or on the PC side], DMC
should detect it and signal the error")
WaitTime(2, "")

#check that the 1355 is not connected anymore
ChkBit(202, 19, 1)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_int_nom.xls")
ChkReport()
```

RES.1: Resource
Script file: a_resource.txt

```
Before the test, connect to SPU simulator to J02 and J03, restart DMC and BOLC
```

```
ChkClearReport ()
```

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Make sure the mechanisms are connected")
WaitForGo("Start 2 Basic receivers to simulate SPUs, configure them to receive data and throw them
away [option 5]")

WaitForGo("Write down the power consumption [DMC OBSW ALONE]");

# Start the link with blue SPU
Trig (87,1,1)

# Start the link with red SPU
Trig (86,1,1)

# Switch on blue DEC and red DEC
Trig (12,0,0)
Trig(19, 0, 0)
WaitTime(10,"")

#Red DEC simulator must be configured to send 256 readouts/sec
Write2(27, 32, 8)
Trig(23, 0, 0)

# 2-10-1-2 to synchronize on blue DEC
Trig (10,1,1)

WaitForGo("Write down the power consumption [DMC WITH ONE DEC ON]");

# Switch on BOLC Sim
WaitForGo("Switch on BOLC Sim")

# reset SMCS2 chip
Trig (89,0,0)
WaitTime (5,"")
# Configure BOLC to send data
Trig (33,1,0x0902002)

# Forward blue DEC data to Blue SPU
Write1 (23,0)

# Forward red DEC data to Red SPU
Write1 (24,0)

# 2-38-0 to switch on grating
Trig(38,0,0)

# Enable grating controller
Trig (40,0,0)

# Switch on chopper
Trig (49,0,0)

# Enable chopper controller
Trig (51,0,0)

# Switch on calibration source 1
Trig (68,0,0)

# Enable calibration source 1 controller
Trig (91,0,0)

# Switch on calibration source 2
Trig (72,0,0)

# Enable calibration source 2 controller
Trig (93,0,0)

# Upload a sequence
#DMC_LOOP , 100
#  DMC_MOVE_GRAT_REL , 117
#  DMC_WAIT , 32
#DMC_END_LOOP , 0
#DMC_END_SEQUENCE , 0
UploadSeq ("gratSteps")

# Set the hk diag list to its maximum size
Write16(13,0x242,0x243,0x244,0x245,0x246,0x247,0x248,0x249,0x250,0x251,0x252,0x253,0x254,0x255,0x256,
0xFFFF)
```

```
# Start hk diag synchronized on blue DEC note, the maximum frequency is 1KHZ but the requirement was
256Hz
Trig (76,1,1)

# Start the sequence
Trig (5,0,0)
WaitTime (10,"")

#check that the CPU workload is lower than 70%
ChkLT(241,700)

WaitForGo("Write down the power consumption [DMC WITH ONE DEC ON, GRATING AND CHOPPER CONTROLLED]");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_resource_nom.xls")

ChkReport ()
```

REL.1: Reliability
Script file: a_rel.txt

```
Before the test, switch-off DMC, disconnect the grating (remove P11 connector, and switch-
on DMC
```

```
ChkClearReport ()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo("Make sure grating [P11] is not connected to DMC")
WaitForGo("Make sure the proto FW is connected to P13")

#switch-on FW Spec
Trig(58, 0, 0)

#Move the FW spec to location B
Trig(64, 1, 1)
WaitTime(20, "")

#check it is in position B
ChkBit(210, 29, 1)

#Move the FW spec to location A
Trig(64, 1, 0)
WaitTime(3, "")

#check that the FW is moving
ChkBit(210, 25, 1)
ChkBit(210, 26, 1)
WaitTime(17, "")

#check it is in position A
ChkBit(210, 28, 1)

#switch-off FW spec
Trig(39, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_rel_nom.xls")

ChkReport ()
```

TRIG.1: Reliability
Script file: a_trig1.txt

```
Before the test, Switch off DMC, reconnect Grating to J11, connect a SPU Sim to red SPU
link (J03) and switch on DMC
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

Log("_________0_tested in IC.1")
Log("_________1_tested in IC.1")
Log("_________2_tested in IC.1")
Log("_________3_tested in IC.1")
Log("_________4______________")

# Upload the sequence "testLabel"
# DMC_LOOP , 5
#   DMC_LOOP , 2
#     DMC_LABEL , 1
#     DMC_WAIT , 80
#     DMC_LABEL , 2
#     DMC_WAIT , 80
#   DMC_END_LOOP , 0
# DMC_END_LOOP , 0
# DMC_END_SEQUENCE , 0
UploadSeq("testLabel")

#switch on blue DEC
Trig (12,0,0)
WaitTime(8,"")

# Synchronize on blue DEC
```

```
Trig (10,1,1)

# 2-5-0 to start the sequence
Trig (5,0,0)

# Wait for 10 seconds,
WaitTime (10,"")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In Hk.xls, check that DMC_SEQ_LABEL is changing from 1 to 2 and back in the
inner loop.")

#abort the sequence
Trig (6,0,0)

Log("_________5_tested in IC.1")
Log("_________6_tested in IC.1")
Log("_________7______________")

# Write and set new time
Write2(0,0x1234,0x56789ABC)
Trig (7,0,0)
WaitTime (3,"")

#check the time has been changed
ChkEQ(226, 0x1234)
ChkEQ(227, 0x56789ABC)

# Write a new time
Write2(0,0x4321,0xCBA98765)
WaitTime (3,"")

#check the time has not changed yet
ChkEQ(226, 0x1234)
ChkEQ(227, 0x56789ABC)

#set the time
Trig (7,0,0)
WaitTime (3,"")

#check the time has been changed
ChkEQ(226, 0x4321)
ChkEQ(227, 0xCBA98765)

Log("_________8______________")

# Set a new OBSID
Trig ( 8,1,0x12345678)
WaitTime(3, "")

#check it has changed
ChkEQ(224, 0x12345678)

# Set a new OBSID
Trig ( 8,1,0x87654321)
WaitTime(3, "")

#check it has changed
ChkEQ(224, 0x87654321)


Log("_________9______________")

# Set a new BBID
Trig(9, 1, 0xFEDCBA98)
WaitTime(3, "")

#check it has changed
ChkEQ(225, 0xFEDCBA98)

# Set a new OBSID
Trig(9, 1, 0x89ABCDEF)
WaitTime(3, "")

#check it has changed
ChkEQ(225, 0x89ABCDEF)
```

```
Log("_________10______________")

WaitForGo("Make sure BOLC sim is connected and switched on")

# Reset SMCS chip 2 and connect to BOLC
Trig (89,0,0)
WaitTime (1,"")

# Configure the BOLC to send science data
Trig (33,1,0x09020002)

#Upload the sequence testWait.seq
# DMC_WAIT(1000)
# DMC_END_SEQUENCE
UploadSeq("testWait")

#2-10-1-1 to synchronize on the Blue DEC
Trig(10,1,1)

#start the sequence and execute it for 10 seconds
Trig (5,0,0)
WaitTime (10,"")
Trig(6, 0, 0)

# Synchronize on the BOLC
Trig (10,1,0x04)

#start the sequence and execute it for 10 seconds
Trig (5,0,0)
WaitTime (10,"")
Trig(6, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', you should see DMC_SEQ_WAIT_IND incrementing by 64 between 2 hk
packets during the first execution of the sequence and by 80 during the second execution.")


Log("_________11______________")

# Change the IRQ frequency
Write6(29, 0, 0, 0, 0x1A, 0x95217CB, 0x2000B)

#write the new timing parameters in the timing FPGA
Trig (11,0,0)
WaitTime (10,"")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'HK.xls' DMC_IRS_CNT shall increment by 8192 between 2 hk packets.")

#go back to nominal spectro parameters
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)

Log("_________12______________")

#switch-on the blue DEC
Trig (12,0,0)
WaitTime(8,"")

# In DMC_DECB_CTRL_ST, bit19 should be 0 (link connected) and bit20=1 (powered on)
ChkBit(201, 19, 0)
ChkBit(201, 20, 1)

Log("_________13______________")

# Switch-off the blue DEC
Trig (13,0,0)
WaitTime (10,"")

# In DMC_DECB_CTRL_ST, bit19 should be 1 (link disconnected) and bit20=0 (powered off)
ChkBit (201,19,1)
ChkBit (201,20,0)

Log("_________14______________")

#switch-on the blue DEC
```

```
Trig (12,0,0)
WaitTime(8, "")

In DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should be 0
ChkBit(291,15,0)
ChkBit(325,15,0)

# Switch-on blue spectro array
Trig (14,0,0)
WaitTime (15,"")

# In DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 0 to 1
ChkBit(291,15,1)
ChkBit(325,15,1)

Log("_________15______________")

# Switch-off blue spectro array
Trig (15,0,0)
WaitTime(15,"")

# in DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 1 to 0
ChkBit(291,15,0)
ChkBit(325,15,0)

Log("_________16______________")

# Switch-on blue spectro array
Trig (14,0,0)
WaitTime (15,"")

# Change the number of clocks per readout (64)
Write1 (27,0x40)
Trig (16,0,0)
WaitTime (10,"")

# Change the number of clocks per readout (32)
Write1 (27,0x20)
Trig (16,0,0)
WaitTime (10,"")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', after the parameters have been changed for the first time
DMC_DECB_REC_PAC should increment by 256 between 2 hk packets.  After the second command,
it should increment by 512.")

# Switch-off blue spectro array
Trig (15,0,0)
WaitTime(15,"")

Log("_________17_tested in PHD.1B")
Log("_________18_tested in PHD.1B")

Log("_________19______________")

#switch-on the Red DEC
Trig (19,0,0)
WaitTime(8,"")

# In DMC_DECR_CTRL_ST, bit19 should be 0 (link connected) and bit20=1 (powered on)
ChkBit(204, 19, 0)
ChkBit(204, 20, 1)

Log("_________20______________")

# Switch-off the Red DEC
Trig (20,0,0)
WaitTime (10,"")

# In DMC_DECR_CTRL_ST, bit19 should be 1 (link disconnected) and bit20=0 (powered off)
ChkBit (204,19,1)
ChkBit (204,20,0)

Log("_________21______________")

#switch-on the Red DEC
Trig (19,0,0)
```

```
WaitTime(8, "")

In DMC_DECR_CR_ST_1 and DMC_DECR_CR_ST_2, bit15 should be 0
ChkBit(359,15,0)
ChkBit(393,15,0)

# Switch-on Red spectro array
Trig (21,0,0)
WaitTime (15,"")

# In DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 0 to 1
ChkBit(359,15,1)
ChkBit(393,15,1)

Log("_________22______________")

# Switch-off blue spectro array
Trig (22,0,0)
WaitTime(15,"")

# in DMC_DECB_CR_ST_3 and DMC_DECB_CR_ST_4, bit15 should change from 1 to 0
ChkBit(359,15,0)
ChkBit(393,15,0)

Log("_________23______________")

# Switch-on red spectro array
Trig (19,0,0)
WaitTime (15,"")

# Change the number of clocks per readout (64)
Write1 (26,0x40)
Trig (23,0,0)
WaitTime (10,"")

# Change the number of clocks per readout (32)
Write1 (26,0x20)
Trig (23,0,0)
WaitTime (10,"")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', after the parameters have been changed for the first time
DMC_DECR_REC_PAC should increment by 256 between 2 hk packets.  After the second command,
it should increment by 512.")



Log("_________24_______________")

#switch-on red DEC
Trig(19, 0, 0)
WaitTime(8, "")

#send the parameters to both spectro array at the same time
Trig (24,0,0)
#Wait 10 sec
WaitTime (10,"")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', check DMC_DECB_RO_CO_3 and DMC_DECR_RO_CO_1.  After the command,
they should always have the same value.")

Log("________25______________")

WaitForGo("On the Blue SPU Sim, start 'Blue Link Receiver ' configured to show if the
science packets are valid or invalid [option 11].")
WaitForGo("Switch-on BOLC Sim")

#start link with blue SPU and forward blue data from blue DEC to blue SPU
Trig(87, 1, 1)
Write1(23, 0)

ManualCheck ("The Link Receiver shall display ‘I’")

#validate blue science data
Trig(25, 0, 0)
ManualCheck ("The Link Receiver shall display ‘V’")
```

```
Log("________28______________")

#invalidate blue science data
Trig(28, 0, 0)
ManualCheck ("The Link Receiver shall display ‘I’")

Log("________27______________")

#validate all science data
Trig(27, 0, 0)
ManualCheck ("The Link Receiver shall display ‘V’")

Log("________30______________")

#invalidate all science data
Trig(30, 0, 0)
ManualCheck ("The Link Receiver shall display ‘I’")

#configure packet encoders such that red SPU data from red DEC goes to blue SPU
Write1(23, 4)
Write1(31, 2)
Write1(24, 0)

ManualCheck ("The Link Receiver shall display ‘I’")

Log("________26______________")

#validate red science data
Trig(26, 0, 0)
ManualCheck ("The Link Receiver shall display ‘V’")

Log("________29______________")

#invalidate red science data
Trig(29, 0, 0)
ManualCheck ("The Link Receiver shall display ‘I’")

Log("________27______________")

#validate all science data
Trig(27, 0, 0)
ManualCheck ("The Link Receiver shall display ‘V’")

Log("________30______________")

#invalidate all science data
Trig(30, 0, 0)
ManualCheck ("The Link Receiver shall display ‘I’")

Log("_________31_tested in PHA.3")
Log("_________32_tested in PHA.3")
Log("_________33_tested in TRIG.25")
Log("_________34_tested in PHD.1a")
Log("_________35_tested in PHD.1a")
Log("_________36_spare")
Log("________37______________")

# Reset the BOLC readout counter
Trig (37,0,0)
WaitTime(4,"")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', DMC_BOL_READ_CNT shall have been reset.")

Log("__grating commands__________")
Log("_________38_tested in GRAT.2")
Log("_________39_tested in GRAT.2")
Log("_________40_tested in GRAT.2")
Log("_________41_tested in GRAT.2")
Log("_________42_tested in GRAT.2")
Log("_________43_tested in GRAT.2")
Log("_________44_tested in GRAT.2")
Log("_________45_tested in GRAT.3")
Log("_________46_tested in GRAT.3")
Log("_________47_tested in GRAT.1")
Log("_________48_tested in GRAT.1")
Log("__chopper commands__________")
Log("_________49_tested in CHOP.1")
Log("_________50_tested in CHOP.1")
Log("_________51_tested in CHOP.1")
Log("_________52_tested in CHOP.1")
Log("_________53_tested in CHOP.1")
Log("_________54_tested in CHOP.1")
Log("_________55_tested in CHOP.1")
Log("_________56_tested in CHOP.1")
Log("_________57_tested in CHOP.2")

Log("__FW commands_____________")
Log("_________58_tested in FW.1")
Log("_________59_tested in FW.1")
Log("_________64_tested in FW.1")
Log("_________65_tested in FW.1")
Log("_________66_tested in FW.1")
Log("_________67_tested in FW.1")

Log("__Blue Heater and Flasher__")
Log("_________60_tested in PHD.1b")
Log("_________61_tested in PHD.1b")
Log("_________62_tested in PHD.1b")
Log("_________63_tested in PHD.1b")

Log("__BB commands_____________")
Log("_________68_tested in BB.1")
Log("_________69_tested in BB.1")
Log("_________70_tested in BB.1")
Log("_________71_tested in BB.2")
Log("_________72_tested in BB.1")
Log("_________73_tested in BB.1")
Log("_________74_tested in BB.1")
Log("_________75_tested in BB.2")

Log("__HK diag commands________")
Log("_________76_tested in HKD.1")
Log("_________77_tested in HKD.2")

Log("_________78_can not be tested [internal command]")

Log("__Red Heater and Flasher__________")
Log("_________79_tested in PHD.1a")
Log("_________80_tested in PHD.1a")
Log("_________81_tested in PHD.1a")
Log("_________82_tested in PHD.1a")

Log("_________83_can not be tested [spare command]")
Log("_________84_can not be tested [spare command]")
Log("_________85_can not be tested [debug command]")

Log("__SPU interface____________")
Log("_________86_tested in PHA.2")
Log("_________87_tested in BOA.1")


Log("________88______________")
#copy the OBS to EEPROM
Trig(88, 0, 0)
WaitTime(8, "")

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck("In DMC_SW_GLOBAL_ST, check that bit18=1 during the copy in EEPROM.")

Log("_________89_tested in BOA.1")
Log("_________90_can not be tested [debug command]")

Log("__BB commands_____________")
Log("_________91_tested in BB.1")
Log("_________92_tested in BB.1")
Log("_________93_tested in BB.1")
Log("_________94_tested in BB.1")


Log("__Temperature sensors_____")
Log("_________95_tested in TS.1")
Log("_________96_tested in TS.1")


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_trig1_nom.xls")
ChkReport()
```

WRT.1: Write commands
Script file: a_write1.txt

```
Start a new test.  Make sure that the mechanisms are connected (including CS simulator)
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

Log("_________128_tested in TRIG.7")
Log("_________129_tested in TRIG.10")
Log("_________130_same as 129, should be tested with real DPU")
Log("_________131_same as 129, should be tested with real DPU")
Log("_________132_same as 129, should be tested with real DPU")
Log("_________133_same as 129, should be tested with real DPU")
Log("_________134_same as 129, should be tested with real DPU")
Log("_________135_same as 129, should be tested with real DPU")
Log("_________136_same as 129, should be tested with real DPU")
Log("_________137_same as 129, should be tested with real DPU")
Log("_________138_same as 129, should be tested with real DPU")
Log("_________139_same as 129, should be tested with real DPU")
Log("_________140_not imlemented")
Log("_________141_tested in HKD.1")

Log("_________142_______________")

#get the value of DMC_CUSTOM_ENT_1
PrintHk(433)

#configure the first custom hk to monitor the content of DRAM at 0x60000 (this address
contains the address where the time is stored)
Write5(14, 0x60000, 1, 4, 0, 1)
WaitTime(3, "")

PrintHk(433)
ManualCheck("Check that the value of DMC_CUSTOM_ENT_1 has changed, meaning it is now
monitoring another value")

Log("_________143_______________")

#switch-on blue DEC and synchronize on it
Trig(12, 0, 0)
WaitTime(5, "")
Trig(10, 1, 1)

# Switch-on grating
Trig (38,0,0)
waitTime (2,"")

#Write STM Grating params
#Write Hall sensors offset
Write1(34, 636);
#Write Range
Write1(33, 0x100000)
#Nominal parameters
Write8(15, 0x1388, 0x3d090, 0x28, 0, 0x3, 0x10e4311, 0x452f, 1631)


# Enable grating controller
Trig (40,0,0)
WaitTime (1,"")

# Home the grating
Trig(44,1,0)
WaitForGo ("Wait that the homing has completed")

# Move it to a central position
Trig (42,1,0x3C000)
WaitTime (15,"")
```

```
# Move it
Trig (43,1,0x3C000)
#Wait 15 seconds
WaitTime (15,"")

# Change the rate (and copy the default parameter of the PID)
Write5 (15,0x3e8,0xc350,0x12,0x0,12)

# Move it
Trig (43,1,0x3c000)
WaitForGo ("Wait that the move has completed")
Write5 (15,0x3e8,0xc350,0x12,0x0,0x3)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'hk.xls', DMC_GRAT_SETPOIN is incrementing 4 times faster during the last
move , after we have changed the Rate from 3 to 12")

Log("_________149_______________")
Write1(21, 1)
Trig(44, 1, 0)

ManualCheck("Check that the homing could not complete because the power limit was too
low");

#write back the default value
Write1(21, 8856)

# Disable grating controller
Trig (41,0,0)

# Switch-off grating
Trig (39,0,0)


Log("_________144_______________")

# Switch-on chopper
Trig (49,0,0)

# Enable chopper controller
Trig (51,0,0)

# change the hk diag list and start diag hk at 1Khz
Write5(13, 244, 245, 258 , 242, 0xFFFF)
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hkdiag.dat")
Trig(76, 1, 0)

# Write default values
Write21(16, 0x404f0, 0x1ea5a74, 0x1a5, 0x3de, 0x148, 0x7FFFFFFF, 0x267e, 0x7fff,
0x7fffffff, 4874, 0x186a0, 0x579, 0x538, 602, 1020, 602, 1143, 326, 0x90, 0xaa820, 0x9c4)

# Move it
Trig (53,1,0x3000)
WaitTime (1,"")

# Move it back
Trig (53,1,0x0)
WaitTime (1,"")

# Change the rate (and keep default parameters)
Write21(16, 0x404f0, 0x1ea5a74, 0x1a5, 0x3de, 0x1, 0x7FFFFFFF, 0x267e, 0x7fff, 0x7fffffff,
4874, 0x186a0, 0x579, 0x538, 602, 1020, 602, 1143, 326, 0x90, 0xaa820, 0x9c4)

# Move it
Trig (53,1,0x3000)

# wait 4 seconds
WaitTime (4,"")

# Move it back
Trig (53,1,0)

#wait 4 seconds
WaitTime (4,"")

# stop hk diag
```

```
Trig(77, 0, 0)
WaitTime(0.5, "")

System("D:\prj\PACS\TestPlanTools\View_HK_diag_chopper.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")

# Disable the controller
Trig (52,0,0)
# Switch-off chopper
Trig (50,0,0)
Write21(16, 0x404f0, 0x1ea5a74, 0x1a5, 0x3de, 0x148, 0x7FFFFFFF, 0x267e, 0x7fff,
0x7fffffff, 4874, 0x186a0, 0x579, 0x538, 602, 1020, 602, 1143, 326, 0x90, 0xaa820, 0x9c4)

ManualCheck ("In 'hkDiag.xls', DMC_CHOP_SETPOIN should increment very fast before the write
commands, and very slowly after.")

Log("_________145_______________")

WaitForGo ("Connect the proto FW to connector J13")

# Switch-on FW Spec
Trig (58,0,0)

# Write default rate
Write1 (17,0x64)

# Make a complete turn
Trig (65,1,0x600)

# wait turn completed
WaitForGo ("Press OK when the first turn is completed ? ")

# Change the rate (and keep default parameters)
Write1 (17,0x32)

# Make a complete turn
Trig (65,1,0x600)

# wait turn completed
WaitForGo ("Press OK when the second turn is completed ? ")

# Switch-off FW Spec
Trig (39,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'hk.xls', in DMC_FW_SPEC_CTRL, you should see in bit25 [moving] that the
FW is moving 2 times faster")

Log("_________146_______________")

WaitForGo ("Connect the proto FW to connector J14")
# Switch-on FW Photo
Trig (59,0,0)

# Write default rate
Write1 (18,0x64)

# Make a complete turn
Trig (67,1,0x600)

# wait turn completed
WaitForGo ("First Turn completed ? ")

# Change the rate (and keep default parameters)
Write1 (18,0x32)
# Make a complete turn
Trig (67,1,0x600)

# wait turn completed
WaitForGo ("Second Turn completed ? ")
# Switch-off FW Photo
Trig (39,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'hk.xls', in DMC_FW_SPEC_CTRL, you should see in bit25 [moving] that the
FW is moving 2 times faster")
```

```
Log("_________147_______________")

#switch on BB1 and enable controller
Trig (68,0,0)
Trig (91,0,0)

#set the target resistor value to 200ohms.
Trig (70, 1, 2000000)
WaitTime(45, "")

PrintHk(446)
ManualCheck("Check that the DMC_CS1_OUTPUT is +/-32767")

#change the Output limit parameter to 16383
Write6 (19,1000000,5000,3277,1857,327,0x3fff)
WaitTime(45, "")

PrintHk(446)
ManualCheck("Check that the DMC_CS1_OUTPUT is +/-16383")

#set default parameters back
Write6 (19,1000000,5000,3277,1857,327,0x7fff)

#disable BB1 controller and switch off
Trig (92,0,0)
Trig (69,0,0)

Log("_________148_______________")

#switch on BB2 and enable controller
Trig (72,0,0)
Trig (93,0,0)

#set the target resistor value to 200ohms.
Trig (74, 1, 2000000)
WaitTime(45, "")

PrintHk(448)
ManualCheck("Check that the DMC_CS2_OUTPUT is +/-32767")

#change the Output limit parameter to 16383
Write6 (20,1000000,5000,3277,1857,327,0x3fff)
WaitTime(45, "")

PrintHk(448)
ManualCheck("Check that the DMC_CS2_OUTPUT is +/-16383")

#set default parameters back
Write6 (20,1000000,5000,3277,1857,327,0x7fff)

#disable BB1 controller and switch off
Trig (94,0,0)
Trig (73,0,0)

Log("_________149_tested before in this test")
Log("_________150_tested in BOA.1")
Log("_________151_tested in RES.1")
Log("_________152_tested in RES.1")

Log("_________153_______________")

# Change the Max Dither value.
Write1 (25,0x20)
WaitTime (5,"")
#In DMC_CHOP_MAX_DIT should go to value 32.
ChkEQ (249,32)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_write1_nom.xls")

ChkReport()
```

WRT.2: Write commands
Script file: a_write2.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

Log("_________154_______________")

#go to spectro mode (timing only)
Write6(29, 0, 0, 0, 26, 0x095217cb, 0xB)
Trig(11, 0, 0)
WaitTime(2.5, "")
Trig(10, 1, 1)

#switch on red DEC
Trig(19, 0, 0)
WaitTime(10, "")

#write default parameters for red DEC
Write2(26, 20, 8)
WaitTime(0.5, "")
Trig(23, 0, 0)
WaitTime(3, "")

#check that the number of readouts/ramp is 8
ChkEQ(358, 8)

#write the new value
Write2(26, 20, 16)
WaitTime(3, "")

#check that the number of readouts/ramp is still 8
ChkEQ(358, 8)

#send the parameters to red DEC
Trig(23, 0, 0)
WaitTime(3, "")

#check that the number of readouts/ramp is now 16
ChkEQ(358, 16)

#switch-off red DEC
Trig(20, 0, 0)

Log("_________155_______________")

#switch on blue DEC
Trig(12, 0, 0)
WaitTime(7, "")

#write default parameters for blue DEC
Write2(27, 20, 8)
WaitTime(0.5, "")
Trig(16, 0, 0)
WaitTime(3, "")

#check that the number of readouts/ramp is 8
ChkEQ(290, 8)

#write the new value
Write2(27, 20, 16)
WaitTime(3, "")

#check that the number of readouts/ramp is still 8
ChkEQ(290, 8)

#send the parameters to blue DEC
Trig(16, 0, 0)
WaitTime(3, "")

#check that the number of readouts/ramp is now 16
ChkEQ(290, 16)

Log("_________156_tested in IC.1")
Log("_________157_tested in TRIG.1")
Log("_________158_tested in TRIG.1")
Log("__________159_tested in TRIG.1")

Log("__________160________________")

#Write STM Grating params
#Write Hall sensors offset
Write1(34, 636);
#Write Range
Write1(33, 0x100000)
#Nominal parameters
Write8(15, 0x1388, 0x3d090, 0x28, 0, 0x3, 0x10e4311, 0x452f, 1631)


#switch-on grating
Trig(38, 0, 0)
Trig(40, 0, 0)

#set the default value for the inductosyn amplitude
Write1(32, 4095)

#start hk diag to monitor the sine and cosine amplitude (DMC_GR_IND_SINE and
DMC_GR_IND_COS)
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")
Write3(13, 539, 540, 0xFFFF)

Trig(76, 1, 50)

Trig(43, 1, 0x30000)
WaitTime(10, "")

#change the amplitude to 50% of the previous one
Write1(32, 2047)

#wait that the change takes effect
WaitTime(60, "inductosyn amplitude is being adjusted")

Trig(43, 1, -0x30000)
WaitTime(10, "")

Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_grat_induct_adjust.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls")

ManualCheck("In HkDiag.xls, Maximum values should have decreased by 50%")

#set the default value for the inductosyn amplitude and wait until it reaches the value
Write1(32, 4095)
WaitTime(60, "")

Log("__________161________________")

#home toward positive position
Trig(44, 1, 1)
WaitForGo("Wait homing completed")
PrintHk(250)


#disable grating and change the range
Trig(41, 0, 0)
Write1(33, 0xA0000)

#home toward positive position
Trig(40, 0, 0)
Trig(44, 1, 1)
WaitForGo("Wait homing completed")
PrintHk(250)

ManualCheck("Check that the mechanical position after the second homing is not the same as
after the first homing [there should be a difference of 4 periods of inductosyn], the
numerical position should be the same")

#one homing in the other direction to record more data
Trig(44, 1, 0)
WaitForGo("Wait homing completed")

#stop hk diag and save diag file for later analysis
Trig(77, 0, 0)
System("D:\prj\PACS\TestPlanTools\View_HK_diag_grat_induct_adjust.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\Debug\hkdiag.xls
D:\prj\PACS\TestAcceptance\a_write1_induct.xls")
System("del d:\prj\pacs\simulators\simdpuv26\asw\hkDiag.dat")

Log("_________162_______________")

#configure hk diag list to monitor the position, the setpoint, the output and the ISR
counter
Write5(13, 0xFA, 0xFB, 0x1C4, 0xF2, 0xFFFF)

#start the viewer
WaitForGo("Launch HkDiagRTViewer")

#start hk diag at 1KHz
Trig(76, 1, 0)

#change the offset.
Write1(34, 100000)
WaitTime(3, "")

#Move the grating.  Since the hall sensors have a completely wrong value, it should not
move correctly
Trig(43, 1, 0x50000)
ManualCheck("Press 'OK' if the grating did not move correctly")

DmcIsAlive()

#write default hall sensor offset again
Write1(34, 3000)

Trig(77, 0, 0)

Log("_________163_______________")

#disable grating controller
Trig(41, 0, 0)
WaitTime(0.5, "")
Trig(39, 0, 0)
WaitTime(0.5, "")

#enter grating open loop mode
Trig(38, 0, 0)
WaitTime(1, "")
Trig(45, 1, 0)

#move the grating
Trig(42, 1, 0x1000)

WaitTime(20, "")

#move the grating relative
Trig(43, 1, 0x1000)
WaitTime(20, "")

#change the rate
Write1(35, 16)

#move the grating relative
Trig(43, 1, -0x1000)
WaitTime(20, "")

#switch off and disable grating
Trig(41, 0, 0)
Trig(39, 0, 0)
ManualCheck("The grating should have moved 2 times faster during the second move")

Log("_________164_______________")

#change the grating controller ouptut filter
Write1(36, 0)

#switch-on grating
Trig(38, 0, 0)
Trig(40, 0, 0)

#move the grating
Trig(44, 1, 0)
ManualCheck("The grating should not move")
```

```
#switch off and disable grating
Trig(41, 0, 0)
Trig(39, 0, 0)

#switch-off blue DEC
Trig(13, 0, 0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_write2_nom.xls")


ChkReport()
```

DATA.1: DEC Data group 1 calibration
Script file: cal_data1.txt

```
Switch-off DMC
Connect the distribution board 1 to the J151 and J153 connector
Connect the CRE output voltage simulator to the J157 and J155 connector and connect REF to
VSS of the distribution board
Switch-on DMC
```

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

# SWON the Red DEC
Trig(19,0,0)

WaitTime(10,"")

# start the CRE
Trig(21,0,0)

WaitTime(20,"")

Write5(26, 20, 64, 0x8F, 82, 0)
Trig(23, 0, 0)
WaitTime(1, "")




# configure packet encoders such that the red SPU data from Red DEC goes to Blue SPU
Write1(23,4)
Write1(31,2)

WaitForGo("Start the Link Receiver to replace the Blue SPU, select Pixel History");

# Start the link with blue SPU
Trig(87,1,1 )


WaitForGo("remove all the jumpers on the DATA_CAL_BOARD");

# Forward the science data from Blue DEC to SPU
Write1(24,0)

WaitForGo("Add the jumper1  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper2  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper3  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

# SWOFF the Red DEC
Trig(20,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_data1_nom.xls")
```

DATA.2: DEC Data group 2 calibration
Script file: cal_data2.txt

```
Switch-off DMC
Connect the distribution board 1 to the J162 and J164 connector
Connect the CRE output voltage simulator to the J166 and J168 connector and connect REF to
VSS of the distribution board
Switch-on DMC
```

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

# SWON the Red DEC
Trig(19,0,0)

WaitTime(10,"")

# start the CRE
Trig(21,0,0)
```

```
WaitTime(20,"")

Write5(26, 20, 64, 0x8F, 82, 0)
Trig(23, 0, 0)
WaitTime(1, "")



# configure packet encoders such that the red SPU data from Red DEC goes to Blue SPU
Write1(23,4)
Write1(31,2)

WaitForGo("Start the Link Receiver to replace the Blue SPU, select Pixel History");

# Start the link with blue SPU
Trig(87,1,1 )


WaitForGo("remove all the jumpers on the DATA_CAL_BOARD");

# Forward the science data from Blue DEC to SPU
Write1(24,0)

WaitForGo("Add the jumper1  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper2  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper3  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

# SWOFF the Red DEC
Trig(20,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_data2_nom.xls")
```

DATA.3: DEC Data group 3 calibration
Script file: cal_data3.txt

```
Switch-off DMC
Connect the distribution board 1 to the J51 and J53 connector
Connect the CRE output voltage simulator to the J57 and J55 connector and connect REF to
VSS of the distribution board
Switch-on DMC
```

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

# SWON the blue DEC
Trig(12,0,0)

WaitTime(10,"")

# start the CRE
Trig(14,0,0)

WaitTime(20,"")

Write5(27, 20, 64, 0x8F, 82, 0)
Trig(16, 0, 0)
WaitTime(1, "")


WaitForGo("Start the Link Receiver to replace the Blue SPU, select Pixel History");

# Start the link with blue SPU
Trig(87,1,1 )

WaitForGo("remove all the jumpers on the DATA_CAL_BOARD");

# Forward the science data from Blue DEC to SPU
Write1(23, 0)
```

```
WaitForGo("Add the jumper1  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper2  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper3  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

# SWOFF the blue DEC
Trig(13,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_data3_nom.xls")
```

DATA.4: DEC Data group 4 calibration
Script file: cal_data4.txt

```
Switch-off DMC
Connect the distribution board 1 to the J62 and J64 connector
Connect the CRE output voltage simulator to the J66 and J68 connector and connect REF to
VSS of the distribution board
Switch-on DMC
```

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

# SWON the blue DEC
Trig(12,0,0)

WaitTime(10,"")

# start the CRE
Trig(14,0,0)

WaitTime(20,"")

Write5(27, 20, 64, 0x8F, 82, 0)
Trig(16, 0, 0)
WaitTime(1, "")


WaitForGo("Start the Link Receiver to replace the Blue SPU, select Pixel History");

# Start the link with blue SPU
Trig(87,1,1 )

WaitForGo("remove all the jumpers on the DATA_CAL_BOARD");

# Forward the science data from Blue DEC to SPU
Write1(23, 0)

WaitForGo("Add the jumper1  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper2  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

WaitForGo("Add the jumper3  on the DATA_CAL_BOARD");
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");

# SWOFF the blue DEC
Trig(13,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cal_data4_nom.xls")
```

SUPPLY.2: DEC Supply1 Noise measurement
Script file: a_supnoise1.txt

```
Connect test panel connectors P51 and P53 [distribution board 2] to SupplyBoard1 J151 and
J153.
In order to measure the noise of the Supply voltages re-inject the CRE supply voltages in
```

```
the data lines by connecting test panel connectors P55 and P57 [distribution board] to
DataBoard1 J155 and J157.

Start a new test
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel connectors P51 and P53 [distribution board] to SupplyBoard1
J151 and J153.");
WaitForGo ("In order to measure the noise of the Supply voltages re-inject the CRE supply
voltages in the data lines by connecting test panel connectors P55 and P57 [distribution
board] to DataBoard1 J155 and J157.");

# SWON the Red DEC
Trig(19,0,0)

WaitTime(10,"")

# start the CRE
Trig(21,0,0)

WaitTime(20,"")

# configure packet encoders such that the red SPU data from Red DEC goes to Blue SPU
Write1(23,4)
Write1(31,2)



WaitForGo("Start the Link Receiver to replace the Blue SPU and select Pixel History[9]");

# Start the link with blue SPU
Trig(87,1,1 )

# Forward the science data from Blue DEC to SPU
Write1(24, 0)


WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")


# SWOFF the Red DEC
Trig(20,0,0)

WaitForGo("Move PixelHistory files in the /Suplly1_Noise Folder");


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_supnoise1_nom.xls")
ChkReport()
```

SUPPLY.3: DEC Supply2 Noise measurement
Script file: a_supnoise2.txt

```
Connect test panel connectors P62 and P64 [distribution board 2] to SupplyBoard2 J162 and
J164.
In order to measure the noise of the Supply voltages re-inject the CRE supply voltages in
the data lines by connecting test panel connectors P66 and P68 [distribution board] to
DataBoard2 J166 and J168.

Start a new test
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel connectors P51 and P53 [distribution board] to SupplyBoard1
J162 and J164.");
WaitForGo ("In order to measure the noise of the Supply voltages re-inject the CRE supply
voltages in the data lines by connecting test panel connectors P55 and P57 [distribution
board] to DataBoard1 J166 and J168.");
```

```
# SWON the Red DEC
Trig(19,0,0)

WaitTime(10,"")

# start the CRE
Trig(21,0,0)

WaitTime(20,"")

# configure packet encoders such that the red SPU data from Red DEC goes to Blue SPU
Write1(23,4)
Write1(31,2)



WaitForGo("Start the Link Receiver to replace the Blue SPU and select Pixel History[9]");

# Start the link with blue SPU
Trig(87,1,1 )

# Forward the science data from Blue DEC to SPU
Write1(24, 0)


WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")


# SWOFF the Red DEC
Trig(20,0,0)

WaitForGo("Move PixelHistory files in the /Suplly1_Noise Folder");


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_supnoise2_nom.xls")
ChkReport()
```

SUPPLY.4: DEC Supply3 Noise measurement
Script file: a_supnoise3.txt

```
Connect test panel connectors P51 and P53 [distribution board 2] to SupplyBoard3 J51 and
J53.
In order to measure the noise of the Supply voltages re-inject the CRE supply voltages in
the data lines by connecting test panel connectors P55 and P57 [distribution board] to
DataBoard3 J55 and J57.

Start a new test
```

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel connectors P51 and P53 [distribution board] to SupplyBoard3
J51 and J53.");
WaitForGo ("In order to measure the noise of the Supply voltages re-inject the CRE supply
voltages in the data lines by connecting test panel connectors P55 and P57 [distribution
board] to DataBoard3 J55 and J57.");

# SWON the blue DEC
Trig(12,0,0)

WaitTime(10,"")

# start the CRE
Trig(14,0,0)

WaitTime(20,"")
```

```
WaitForGo("Start the Link Receiver to replace the Blue SPU and select Pixel History[9]");

# Start the link with blue SPU
Trig(87,1,1 )

# Forward the science data from Blue DEC to SPU
Write1(23, 0)


WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")


# SWOFF the blue DEC
Trig(13,0,0)

WaitForGo("Move PixelHistory files in the /Suplly3_Noise Folder");


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_supnoise3_nom.xls")
ChkReport()
```

SUPPLY.5: DEC Supply4 Noise measurement
Script file: a_supnoise4.txt

Connect test panel connectors P62 and P64 [distribution board 2] to SupplyBoard4 J62 and J64.
In order to measure the noise of the Supply voltages re-inject the CRE supply voltages in the data lines by connecting test panel connectors P66 and P68 [distribution board] to DataBoard4 J66 and J68.

Start a new test

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel connectors P62 and P64 [distribution board] to SupplyBoard4
J62 and J64.");
WaitForGo ("In order to measure the noise of the Supply voltages, re-inject the CRE supply
voltages in the data lines by connecting test panel connectors P66 and P68 [distribution
board] to DataBoard4 J66 and J68.");

# SWON the blue DEC
Trig(12,0,0)

WaitTime(10,"")

# start the CRE
Trig(14,0,0)

WaitTime(20,"")

WaitForGo("Start the Link Receiver to replace the Blue SPU and select Pixel History[9]");

# Start the link with blue SPU
Trig(87,1,1 )

# Forward the science data from Blue DEC to SPU
Write1(23, 0)


WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")
```

```
WaitForGo("Record 256 readouts.  Press a key on SPU Sim");
WaitTime(3,"")

# SWOFF the blue DEC
Trig(13,0,0)

WaitForGo("Move PixelHistory files in the /Suplly4_Noise Folder");

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_supnoise4_nom.xls")
ChkReport()
```

CRE.1: DEC Supply Group 3-4 verification
Script file: a_cre_intfB.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel 'distribution board' to supply board 3")

# Switch-on Blue DEC
Trig (12,0,0)

#Wait 10 sec
WaitTime (20,"")


#Check that DMC_DECB_VWELL_3  = 0V [32767d]  +/- 2mV
ChkGT (276,32701)
ChkLT (276,32833)


# Check that DMC_DECB_V0V_3   =  0V [32767d]  +/- 2mV
ChkGT (272,32701)
ChkLT (272,32833)


# Check that DMC_DECB_VDDD_3,  DMC_DECB_VSS_3,   DMC_DECB_VGND_3, DMC_DECB_VCAN1_3
,DMC_DECB_VCAN2_3
#            DMC_DECB_VBIAS_3, DMC_DECB_VBI_R_3, DMC_DECB_VSCP_3, DMC_DECB_VDDR_3,
DMC_DECB_VDDA_3
# are close to 0V [32767d]  +/- 2mV

ChkGT (265,32701)
ChkLT (265,32833)
ChkGT (266,32701)
ChkLT (266,32833)
ChkGT (267,32701)
ChkLT (267,32833)
ChkGT (268,32701)
ChkLT (268,32833)
ChkGT (269,32701)
ChkLT (269,32833)
ChkGT (270,32701)
ChkLT (270,32833)
ChkGT (271,32701)
ChkLT (271,32833)
ChkGT (273,32701)
ChkLT (273,32833)
ChkGT (274,32701)
ChkLT (274,32833)
ChkGT (275,32701)
ChkLT (275,32833)

# Switch on blue detector array
Trig (14,0,0)

# Wait 20 sec
WaitTime (20,"")

# Check that DMC_DECB_VDDD_3  is close to 2.5V(46420) +/- 100 mV ")
ChkGT (265,45874)
ChkLT (265,46966)
# Check that DMC_DECB_VSS_3   is close to  -3V(16383) +/- 100 mV")
ChkGT (266,15837)
ChkLT (266,16929)
# Check that DMC_DECB_VGND_3  is close to   0V(32767) +/- 100 mV")
ChkGT (267,32221)
ChkLT (267,33313)
# Check that DMC_DECB_VCAN1_3 is close to 0.5V(35496) +/- 100 mV")
ChkGT (268,34950)
ChkLT (268,36042)
# Check that DMC_DECB_VCAN2_3 is close to 1.9V(43143) +/- 100 mV")
ChkGT (269,42597)
ChkLT (269,43689)
# Check that DMC_DECB_VBIAS_3 is close to   0V(32767) +/-   2mV")
ChkGT (270,32701)
ChkLT (270,32833)
```

```
# Check that DMC_DECB_VBI_R_3 is close to   0V(32767) +/-   2mV")
ChkGT (271,32701)
ChkLT (271,32833)
# Check that DMC_DECB_V0V_3   is close to   0V(32767) +/- 100mV")
ChkGT (272,32221)
ChkLT (272,33313)
# Check that DMC_DECB_VSCP_3  is close to-0.1V(32221) +/- 100mV)
ChkGT (273,31675)
ChkLT (273,32767)
# Check that DMC_DECB_VDDR_3  is close to 1.2V(39321) +/- 100mV")
ChkGT (274,38775)
ChkLT (274,39867)
# Check that DMC_DECB_VDDA_3  is close to 2.5V(46420) +/- 100mV")
ChkGT (275,45874)
ChkLT (275,46966)
# Check that DMC_DECB_VWELL_3 is close to 2.5V(46420) +/- 100mV")
ChkGT (276,45874)
ChkLT (276,46966)

# Change Bias D = 1V
Write5 (27,0x20,0x8,0x18c,0x0,0xfff)

# Send the parameter set
Trig (16,0,0)
#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V ")

# Change Bias R = 1V
Write5 (27,0x20,0x8,0x18c,0xfff,0xfff)

# Send the parameter set
Trig (16,0,0)

#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V and between 0BIAS &
BIASR is 1V ")


# Reset biases and change select lines
Write5 (27,0x20,0x8,0x183,0x0,0x0)

#2-16-0 to send the parameter set
Trig (16,0,0)

ManualCheck ("SELECT = VDDD, SEL1 = VSS, SEL2 = VSS")

WaitForGo ("Connect a 50K resistor between VDDA and VSS")

#Wait 4 sec
WaitTime (4,"")

# Check that IDDA_3 is close to 0.11 mA +/- 3% (HK close to 38773)
ChkGT (277,38592)
ChkLT (277,38953)

WaitForGo ("Connect the 50K resistor between VDDD and VSS")
#Wait 4 sec
WaitTime (4,"")

# Check that IDDD_3 = 0.11 mA +/- 3% (HK close to 38773) and ISS_3 =-0.11mA +/- 3% (HK
close to 26578)
ChkGT (278,38592)
ChkLT (278,38953)
ChkGT (279,26578)
ChkLT (279,26939)

# Switch-oFF CREs
Trig (15,0,0)

#Wait 20 sec
WaitTime (20,"")

# Switch-oFF Blue DEC
Trig (13,0,0)
```

```
###########################################################################################
#####


WaitForGo ("Connect test panel 'distribution board' to supply board 4")

# Switch-on Blue DEC
Trig (12,0,0)

#Wait 10 sec
WaitTime (20,"")


#Check that DMC_DECB_VWELL_4  = 0V [32767d]  +/- 2mV
ChkGT (310,32701)
ChkLT (310,32833)


# Check that DMC_DECB_V0V_4   =  0V [32767d]  +/- 2mV
ChkGT (306,32701)
ChkLT (306,32833)

# Check that DMC_DECB_VDDD_4,  DMC_DECB_VSS_4,   DMC_DECB_VGND_4, DMC_DECB_VCAN1_4
,DMC_DECB_VCAN2_4
#            DMC_DECB_VBIAS_4, DMC_DECB_VBI_R_4, DMC_DECB_VSCP_4, DMC_DECB_VDDR_4,
DMC_DECB_VDDA_4
# are close to 0V [32767d]  +/- 2mV

ChkGT (299,32701)
ChkLT (299,32833)
ChkGT (300,32701)
ChkLT (300,32833)
ChkGT (301,32701)
ChkLT (301,32833)
ChkGT (302,32701)
ChkLT (302,32833)
ChkGT (303,32701)
ChkLT (303,32833)
ChkGT (304,32701)
ChkLT (304,32833)
ChkGT (305,32701)
ChkLT (305,32833)
ChkGT (307,32701)
ChkLT (307,32833)
ChkGT (308,32701)
ChkLT (308,32833)
ChkGT (309,32701)
ChkLT (309,32833)



# Switch on blue detector array
Trig (14,0,0)

# Wait 20 sec
WaitTime (20,"")

# Check that DMC_DECB_VDDD_4  is close to 2.5V(46420) +/- 100 mV ")
ChkGT (299,45874)
ChkLT (299,46966)
# Check that DMC_DECB_VSS_4   is close to  -3V(16383) +/- 100 mV")
ChkGT (300,15837)
ChkLT (300,16929)
# Check that DMC_DECB_VGND_4  is close to   0V(32767) +/- 100 mV")
ChkGT (301,32221)
ChkLT (301,33313)
# Check that DMC_DECB_VCAN1_4 is close to 0.5V(35496) +/- 100 mV")
ChkGT (302,34950)
ChkLT (302,36042)
# Check that DMC_DECB_VCAN2_4 is close to 1.9V(43143) +/- 100 mV")
ChkGT (303,42597)
ChkLT (303,43689)
# Check that DMC_DECB_VBIAS_4 is close to   0V(32767) +/-   2mV")
ChkGT (304,32701)
ChkLT (304,32833)
```

```
# Check that DMC_DECB_VBI_R_4 is close to   0V(32767) +/-   2mV")
ChkGT (305,32701)
ChkLT (305,32833)
# Check that DMC_DECB_V0V_4   is close to   0V(32767) +/- 100mV")
ChkGT (306,32221)
ChkLT (306,33313)
# Check that DMC_DECB_VSCP_4  is close to-0.1V(32221) +/- 100mV)
ChkGT (307,31675)
ChkLT (307,32767)
# Check that DMC_DECB_VDDR_4  is close to 1.2V(39321) +/- 100mV")
ChkGT (308,38775)
ChkLT (308,39867)
# Check that DMC_DECB_VDDA_4  is close to 2.5V(46420) +/- 100mV")
ChkGT (309,45874)
ChkLT (309,46966)
# Check that DMC_DECB_VWELL_4 is close to   2.5V(46420) +/- 100mV")
ChkGT (310,45874)
ChkLT (310,46966)

# Change Bias D = 1V
Write5 (27,0x20,0x8,0x18c,0x0,0xfff)

# Send the parameter set
Trig (16,0,0)
#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between OBIAS and GND is 1V ")

# Change Bias R = 1V
Write5 (27,0x20,0x8,0x18c,0xfff,0xfff)

# Send the parameter set
Trig (16,0,0)

#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between OBIAS and GND is 1V and between OBIAS &
BIASR is 1V ")


# Reset biases and change select lines
Write5 (27,0x20,0x8,0x183,0x0,0x0)

#2-16-0 to send the parameter set
Trig (16,0,0)

ManualCheck ("SELECT = VDDD, SEL1 = VSS, SEL2 = VSS")

WaitForGo ("Connect a 50K resistor between VDDA and VSS")

#Wait 4 sec
WaitTime (4,"")

# Check that IDDA_4 is close to 0.11 mA +/- 3% (HK close to 38773)
ChkGT (311,38592)
ChkLT (311,38953)

WaitForGo ("Connect the 50K resistor between VDDD and VSS")
#Wait 4 sec
WaitTime (4,"")

# Check that IDDD_4 = 0.11 mA +/- 3% (HK close to 38773) and ISS_4 =-0.11mA +/- 3% (HK
close to 26578)
ChkGT (312,38592)
ChkLT (312,38953)
ChkGT (313,26578)
ChkLT (313,26939)

# Switch-oFF CREs
Trig (15,0,0)

#Wait 20 sec
WaitTime (20,"")

# Switch-oFF Blue DEC
Trig (13,0,0)
```

```
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cre_intfB_nom.xls")
ChkReport()
```

CRE.2: DEC Supply Group 1-2 verification
Script file: a_cre_intfR.txt

```
ChkClearReport()
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

WaitForGo ("Connect test panel 'distribution board' to supply board 1")

# Switch-on Red DEC
Trig (19,0,0)

#Wait 10 sec
WaitTime (20,"")


#Check that DMC_DECR_VWELL_1  = 0V [32767d]  +/- 2mV
ChkGT (344,32701)
ChkLT (344,32833)


# Check that DMC_DECR_V0V_1   =  0V [32767d]  +/- 2mV
ChkGT (340,32701)
ChkLT (340,32833)


# Check that DMC_DECR_VDDD_1,  DMC_DECR_VSS_1,   DMC_DECR_VGND_1, DMC_DECR_VCAN1_1
,DMC_DECR_VCAN2_1
#            DMC_DECR_VBIAS_1, DMC_DECR_VBI_R_1, DMC_DECR_VSCP_1, DMC_DECR_VDDR_1,
DMC_DECR_VDDA_1
# are close to 0V [32767d]  +/- 2mV

ChkGT (333,32701)
ChkLT (333,32833)
ChkGT (334,32701)
ChkLT (334,32833)
ChkGT (335,32701)
ChkLT (335,32833)
ChkGT (336,32701)
ChkLT (336,32833)
ChkGT (337,32701)
ChkLT (337,32833)
ChkGT (338,32701)
ChkLT (338,32833)
ChkGT (339,32701)
ChkLT (339,32833)
ChkGT (341,32701)
ChkLT (341,32833)
ChkGT (342,32701)
ChkLT (342,32833)
ChkGT (343,32701)
ChkLT (343,32833)

# Switch on Red detector array
Trig (21,0,0)

# Wait 20 sec
WaitTime (20,"")

# Check that DMC_DECR_VDDD_1  is close to 2.5V(46420) +/- 100 mV ")
ChkGT (333,45874)
ChkLT (333,46966)
# Check that DMC_DECR_VSS_1   is close to  -3V(16383) +/- 100 mV")
ChkGT (334,15837)
ChkLT (334,16929)
# Check that DMC_DECR_VGND_1  is close to   0V(32767) +/- 100 mV")
ChkGT (335,32221)
ChkLT (335,33313)
# Check that DMC_DECR_VCAN1_1 is close to 0.5V(35496) +/- 100 mV")
ChkGT (336,34950)
ChkLT (336,36042)
# Check that DMC_DECR_VCAN2_1 is close to 1.9V(43143) +/- 100 mV")
```

```
ChkGT (337,42597)
ChkLT (337,43689)
# Check that DMC_DECR_VBIAS_1 is close to   0V(32767) +/-   2mV")
ChkGT (338,32701)
ChkLT (338,32833)
# Check that DMC_DECR_VBI_R_1 is close to   0V(32767) +/-   2mV")
ChkGT (339,32701)
ChkLT (339,32833)
# Check that DMC_DECR_V0V_1   is close to   0V(32767) +/- 100mV")
ChkGT (340,32221)
ChkLT (340,33313)
# Check that DMC_DECR_VSCP_1  is close to-0.1V(32221) +/- 100mV)
ChkGT (341,31675)
ChkLT (341,32767)
# Check that DMC_DECR_VDDR_1  is close to 1.2V(39321) +/- 100mV")
ChkGT (342,38775)
ChkLT (342,39867)
# Check that DMC_DECR_VDDA_1  is close to 2.5V(46420) +/- 100mV")
ChkGT (343,45874)
ChkLT (343,46966)
# Check that DMC_DECR_VWELL_1 is close to 2.5V(46420) +/- 100mV")
ChkGT (344,45874)
ChkLT (344,46966)

# Change Bias D = 1V
Write5 (26,0x20,0x8,0x18c,0x0,0xfff)

# Send the parameter set
Trig (23,0,0)
#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V ")

# Change Bias R = 1V
Write5 (26,0x20,0x8,0x18c,0xfff,0xfff)

# Send the parameter set
Trig (23,0,0)

#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V and between 0BIAS &
BIASR is 1V ")


# Reset biases and change select lines
Write5 (26,0x20,0x8,0x183,0x0,0x0)

#2-16-0 to send the parameter set
Trig (23,0,0)

ManualCheck ("SELECT = VDDD, SEL1 = VSS, SEL2 = VSS")

WaitForGo ("Connect a 50K resistor between VDDA and VSS")

#Wait 4 sec
WaitTime (4,"")

# Check that IDDA_1 is close to 0.11 mA +/- 3% (HK close to 38773)
ChkGT (345,38592)
ChkLT (345,38953)

WaitForGo ("Connect the 50K resistor between VDDD and VSS")
#Wait 4 sec
WaitTime (4,"")

# Check that IDDD_1 = 0.11 mA +/- 3% (HK close to 38773) and ISS_1 =-0.11mA +/- 3% (HK
close to 26578)
ChkGT (346,38592)
ChkLT (346,38953)
ChkGT (347,26578)
ChkLT (347,26939)

# Switch-oFF CREs
Trig (22,0,0)
```

```
#Wait 20 sec
WaitTime (20,"")

# Switch-oFF Red DEC
Trig (20,0,0)


##############################################################################################
#####



WaitForGo ("Connect test panel 'distribution board' to supply board 4")

# Switch-on Red DEC
Trig (19,0,0)

#Wait 10 sec
WaitTime (20,"")


#Check that DMC_DECR_VWELL_2  = 0V [32767d]  +/- 2mV
ChkGT (378,32701)
ChkLT (378,32833)


# Check that DMC_DECR_V0V_2   =  0V [32767d]  +/- 2mV
ChkGT (374,32701)
ChkLT (374,32833)

# Check that DMC_DECR_VDDD_2,  DMC_DECR_VSS_2,   DMC_DECR_VGND_2, DMC_DECR_VCAN1_2
,DMC_DECR_VCAN2_2
#            DMC_DECR_VBIAS_2, DMC_DECR_VBI_R_2, DMC_DECR_VSCP_2, DMC_DECR_VDDR_2,
DMC_DECR_VDDA_2
# are close to 0V [32767d]  +/- 2mV

ChkGT (367,32701)
ChkLT (367,32833)
ChkGT (368,32701)
ChkLT (368,32833)
ChkGT (369,32701)
ChkLT (369,32833)
ChkGT (370,32701)
ChkLT (370,32833)
ChkGT (371,32701)
ChkLT (371,32833)
ChkGT (372,32701)
ChkLT (372,32833)
ChkGT (373,32701)
ChkLT (373,32833)
ChkGT (375,32701)
ChkLT (375,32833)
ChkGT (376,32701)
ChkLT (376,32833)
ChkGT (377,32701)
ChkLT (377,32833)



# Switch on Red detector array
Trig (21,0,0)

# Wait 20 sec
WaitTime (20,"")

# Check that DMC_DECR_VDDD_2  is close to 2.5V(46420) +/- 100 mV ")
ChkGT (367,45874)
ChkLT (367,46966)
# Check that DMC_DECR_VSS_2   is close to  -3V(16383) +/- 100 mV")
ChkGT (368,15837)
ChkLT (368,16929)
# Check that DMC_DECR_VGND_2  is close to   0V(32767) +/- 100 mV")
ChkGT (369,32221)
ChkLT (369,33313)
# Check that DMC_DECR_VCAN1_2 is close to 0.5V(35496) +/- 100 mV")
ChkGT (370,34950)
ChkLT (370,36042)
# Check that DMC_DECR_VCAN2_2 is close to 1.9V(43143) +/- 100 mV")
```

```
ChkGT (371,42597)
ChkLT (371,43689)
# Check that DMC_DECR_VBIAS_2 is close to   0V(32767) +/-   2mV")
ChkGT (372,32701)
ChkLT (372,32833)
# Check that DMC_DECR_VBI_R_2 is close to   0V(32767) +/-   2mV")
ChkGT (373,32701)
ChkLT (373,32833)
# Check that DMC_DECR_V0V_2   is close to   0V(32767) +/- 100mV")
ChkGT (374,32221)
ChkLT (374,33313)
# Check that DMC_DECR_VSCP_2  is close to-0.1V(32221) +/- 100mV)
ChkGT (375,31675)
ChkLT (375,32767)
# Check that DMC_DECR_VDDR_2  is close to 1.2V(39321) +/- 100mV")
ChkGT (376,38775)
ChkLT (376,39867)
# Check that DMC_DECR_VDDA_2  is close to 2.5V(46420) +/- 100mV")
ChkGT (377,45874)
ChkLT (377,46966)
# Check that DMC_DECR_VWELL_2 is close to   2.5V(46420) +/- 100mV")
ChkGT (378,45874)
ChkLT (378,46966)

# Change Bias D = 1V
Write5 (26,0x20,0x8,0x18c,0x0,0xfff)

# Send the parameter set
Trig (23,0,0)
#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V ")

# Change Bias R = 1V
Write5 (26,0x20,0x8,0x18c,0xfff,0xfff)

# Send the parameter set
Trig (23,0,0)

#Wait 20 sec
WaitTime (20,"")

ManualCheck ("Check that Measure voltage between 0BIAS and GND is 1V and between 0BIAS &
BIASR is 1V ")


# Reset biases and change select lines
Write5 (26,0x20,0x8,0x183,0x0,0x0)

#2-16-0 to send the parameter set
Trig (23,0,0)

ManualCheck ("SELECT = VDDD, SEL1 = VSS, SEL2 = VSS")

WaitForGo ("Connect a 50K resistor between VDDA and VSS")

#Wait 4 sec
WaitTime (4,"")

# Check that IDDA_2 is close to 0.11 mA +/- 3% (HK close to 38773)
ChkGT (379,38592)
ChkLT (379,38953)

WaitForGo ("Connect the 50K resistor between VDDD and VSS")
#Wait 4 sec
WaitTime (4,"")

# Check that IDDD_2 = 0.11 mA +/- 3% (HK close to 38773) and ISS_2 =-0.11mA +/- 3% (HK
close to 26578)
ChkGT (380,38592)
ChkLT (380,38953)
ChkGT (381,26578)
ChkLT (381,26939)

# Switch-oFF CREs
Trig (22,0,0)
```

```
#Wait 20 sec
WaitTime (20,"")

# Switch-oFF Red DEC
Trig (20,0,0)

System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_cre_intfR_nom.xls")
ChkReport()
```

SPUHK.1: SPU housekeeping test
Script file: a_spu_hk.txt

```
ChkClearReport()
WaitForGo("Connect SPU HK simulator")

WaitTime(5, "")
ChkGT(420, -400)
ChkLT(420, -320)

ChkGT(421, -1680)
ChkLT(421, -1600)

ChkGT(422, -16875)
ChkLT(422, -15892)

ChkGT(423, -20)
ChkLT(423, 20)

ChkGT(424, -20)
ChkLT(424, 20)

ChkGT(425, -20)
ChkLT(425, 20)

ChkGT(419, -20)
ChkLT(419, 20)

WaitForGo("Apply 75 mV between pin 4+ and 12- of test connector")
ChkLT(423, 25235)
ChkGT(423, 23735)

WaitForGo("Apply 75 mV between pin 6+ and 14- of test connector")
ChkLT(425, 24937)
ChkGT(425, 23484)

WaitForGo("Apply 3 V between pin 5+ and 13- of test connector")
ChkGT(424, 9536)
ChkLT(424, 10124)

WaitForGo("Apply 4 V between pin 7+ and 15- of test connector")
ChkGT(419, 12714)
ChkLT(419, 13500)

WaitTime(3, "")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_spu_hk_nom.xls")
ChkReport()
```

SF.1: Short Functional test
Script file: a_shortFunc.txt

```
System("del d:\prj\pacs\simulators\simdpuv26\ASW\hk.dat")

#switch-on DECs
Trig (19,0,0)
Trig (12,0,0)
WaitTime (10,"")

#set the readout frequency to 256Hz
Write2 (27,0x20, 8)
```

```
Write2 (26,0x20, 8)
WaitTime(0.5, "")

#send the param to both DEC
Trig (24,0,0)

WaitTime (10,"")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', DMC_DECR_REC_PAC and DMC_DECB_REC_PAC shall increment by 512
between 2 hk packets")

WaitForGo ("Switch on BOLC Sim")
Trig (89,0,0)
WaitTime(2,"")
Trig(33, 1, 0x09020002)
WaitTime(0.5,"")
Trig(33, 1, 0x0B020000)

WaitTime (10,"")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', DMC_BOL_REC_PAC shall increment by 480 between 2 hk packets")

# With Sequence Writer create the sequence:
# DMC_LOOP(100)
#    DMC_MOVE_GRAT_REL(117)
#    DMC_WAIT(32)
# DMC_END_LOOP
# DMC_END_SEQUENCE
# Save it in d:\prj\pacs\sequences\gratSteps.seq
# 6-60-gratSteps to upload the sequence
UploadSeq("gratSteps")

#switch on the grating
Trig (38,0,0)

#change the hk diag list
Write5 (13,0xFA,0xFB,0x1C4,0xF2,0xFFFF)

WaitForGo ("Launch HkDiagRTViewer, press Start, select 'chopperPos.xls'")
Trig (76,1,0x05)

#enable the grating controller
Trig (40,0,0)

#synchronize on blue DEC
Trig (10,1,0x01)

#home the grating
Trig (44,1,0x00)

WaitForGo ("Grating homing finished ?")

#start the sequence
Trig (5,0,0)
WaitForGo ("HkDiagRTViewer shall display the grating response")

# At the end of the sequence:
WaitForGo ("Finish sequence?")
Trig(6, 0, 0)

#stop hk diag
Trig (77,0,0)

#disable the grating controller
Trig (41,0,0)
#switch-off the grating
Trig (39,0,0)

# With Sequence Writer create the sequence:
# DMC_LOOP(100)
#    DMC_MOVE_CHOP_ABS(13434)
#    DMC_WAIT(3)
#    DMC_MOVE_CHOP_ABS(-13434)
#    DMC_WAIT(3)
# DMC_END_LOOP
# DMC_END_SEQUENCE
# Save it in d:\prj\pacs\sequences\simpleChop.seq
# 6-60-simpleChop to upload the sequence

UploadSeq("simpleChop")
#change the hk diag list
Write5 (13,0xF4,0xF5,0x102,0xF2,0xFFFF)


#start diag hk at 1Khz
Trig (76,1,0x00)

#switch on the chopper
Trig (49,0,0)
#enable the chopper controller
Trig (51,0,0)

#synchronize on BOLC
Trig (10,1,0x04)

#start the sequence
Trig (5,0,0)
WaitForGo ("HkDiagRTViewer shall display the chopper response")

# At the end of the sequence:
WaitForGo ("Finish sequence?")
Trig(6, 0, 0)

#stop hk diag
Trig (77,0,0)

#disable the chopper controller
Trig (52,0,0)
#switch off the chopper
Trig (50,0,0)

WaitForGo("Make sure the proto filter wheel is connected to J13")
#switch-on the spectro filter wheel
Trig (58,0,0)
WaitTime (5,"")

#Move the wheel to position A
Trig (64,1,0x00)
WaitForGo ("Is the wheel at position A ?")
#move it to position B
Trig (64,1,0x01)

WaitTime (3,"")
ChkBit (210,25,1)
ChkBit (210,27,1)

WaitForGo ("Wheel at position B ? ")
ChkBit (210,29,1)

#switch-off the spectro filter wheel (DMC_SWOF_GRATING)
Trig (39,0,0)

WaitForGo("Make sure the proto filter wheel is connected to J14")
#switch-on the photo filter wheel
Trig (59,0,0)
WaitTime (5,"")

#Move the wheel to position A
Trig (66,1,0x00)
WaitForGo ("Is the wheel at position A ?")
#move it to position B
Trig (66,1,0x01)

WaitTime (3,"")
ChkBit (211,25,1)
ChkBit (211,27,1)

WaitForGo ("Wheel at position B ? ")
ChkBit (211,29,1)

#switch-off the spectro filter wheel (DMC_SWOF_GRATING)
Trig (39,0,0)
```

```
WaitForGo ("Connect the BB1 cables to a variable Resistor [choose a value between 10 & 100
Ohms]")
#switch on the BB1
Trig (68,0,0)
#enable the BB1 controller
Trig (91,0,0)

WaitTime (30,"")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', check that DMC_CS1_RES_VALUE Is updated with the resistor
value")

#switch off the BB1
Trig (69,0,0)

WaitForGo ("Connect the BB2 cables to a variable Resistor [choose a value between 10 & 100
Ohms]")
#switch on the BB2
Trig (72,0,0)

WaitTime (30,"")
System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("start D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls")
ManualCheck ("In 'Hk.xls', check that DMC_CS2_RES_VALUE Is updated with the resistor
value")

#switch off the BB2
Trig (73,0,0)


System("D:\prj\PACS\TestPlanTools\View_HK.bat")
System("copy D:\prj\PACS\Simulators\SimDPUv26\ASW\hk.xls
D:\prj\PACS\TestAcceptance\a_shortFunc_nom.xls")
```