SOLAR ORBITER SWA
PAS

PAS-DPU ICD

Document under Configuration Control X

Prepared by	**Date**	**Signature:**
Wilfried Marty		Marty Wilfried 2017.09.12 11:28:45 +02'00'
Andrey Fedorov		Digitally signed by Andrey Fedorov DN: cn=Andrey Fedorov, o=IRAP-UPS-CNRS, ou, email=Andrei.Fedorov@irap.omp.eu, c=FR Date: 2017.09.12 10:40:29 +02'00'
Verified by		**Signature:**
Philippe Louarn	Philippe LOUARN	Signature numérique de Philippe LOUARN DN : cn=Philippe LOUARN, o, ou=IRAP, email=philippe.louarn@irap.omp.eu, c=FR Date : 2017.09.12 12:55:14 +02'00'
Mathieu Petiot	Mathieu PETIOT	Signature numérique de Mathieu PETIOT DN : cn=Mathieu PETIOT, o=ALTEN SO, ou=IRAP, email=mathieu.petiot@irap.omp.eu, c=FR Date : 2017.09.12 11:16:43 +02'00'
Sandra Bordon	NEXEYA	Signature numérique de BORDON DN : cn=BORDON, o=Nexeya CF, ou, email=sandra.bordon@nexeya.com, c=FR Date : 2017.09.12 11:00:25 +02'00'
Approved and Application authorized by:		**Signature:**
Andrey Fedorov		Digitally signed by Andrey Fedorov DN: cn=Andrey Fedorov, o=IRAP-UPS-CNRS, ou, email=Andrei.Fedorov@irap.omp.eu, c=FR Date: 2017.09.12 10:40:51 +02'00'

Summary : PAS –DPU Interface control document

Keywords : Interface, tm/tc

Annex:

I DISTRIBUTION LIST

PROJECT	EXTERNAL DISTRIBUTION
Philippe Louarn	Barry Hancock
Andrey Fedorov	Chris Brockley-Blatt
Henry Claude Seran	Chris Owen
Carine Amoros	
Mathieu Petiot	
Jean Rubiella Romeo	Roberto Bruno
Philippe Rouger	Mele Gennaro
Sandra Bordon	
Catherine Garat	
Guillaume Orttner	Lubomir Prech
Rituparna Baruah	
Romain Mathon	

II LIST OF ABBREVIATIONS

SWA	Solar Wind Plasma Analyzer
HIS	Heavy Ion Sensor
PAS	Proton Alpha Sensor
TBC	To Be Completed
TBD	To Be Defined
DPU	Data Processing Unit
DMS	Solar Orbiter Data Management Subsystem
SO	Solar Orbiter spacecraft

III DOCUMENT CHANGE RECORD

Edition	Revision	Date	Modified pages	Reason for change / Observations
1	0	10/02/2014	/	First Issue
2	0	23/06/2015	9	Add a short description of the PAS functions
2	0	15/10/2015	15	A121 enable/disable table reversed
2	0	15/10/2015	15	Scientific header format is updated
2	0	04/11/2015	All	Important document update
2	0	04/11/2015	All	Requirements numbering is changed
2	1	04/02/2016	All	Corrections after the DRAFT review
2	1	05/02/2016	20-21	Table 3, Scientific Header format
2	2	30/09/2016	40	Table 23. New Layout.
2	2	30/09/2016	10-11	Small corrections
2	2	30/09/2016	33	Table 17. New variables
2	2	30/09/2016	16	Table 2. HK packet
2	2	30/09/2016	41-48	Table 22,23,24, section 10.1, 9.5.11, 9.5.12, 9.5.15, Sequencer configuration tables can be updated in IDLE mode ONLY.
2	2	30/09/2016	54	New Section 11
2	2	30/09/2016	11-12	Figure 2 and section 3.3.4 edited, maximal packet size.
2	2	30/09/2016	38	Table 17
2	2	30/09/2016	23-25	Tables 6 and 7. FDIR actions are modified
2	2	30/09/2016	30	Tables 8 and requirement 063-001 are modified
2	2	30/09/2016	30	Data products are modified

2	2	30/09/2016	32	Table 11, format of data
2	2	30/09/2016	34	Table 13 insert
2	2	30/09/2016	34	Figure 5 insert
2	3	14/11/2016	34	DPU-RPW communication is an option
2	3	14/11/2016	16	PAS-ICD-051-001 is modified
2	3	14/11/2016	42	getSequencer command
2	3	14/11/2016	48	Table 25, sharedMemoryDump
2	3	14/11/2016	51	9.5.16. sharedMemoryDump command description
2	3	14/11/2016	52	9.5.15 loadPulseTable
2	3	14/11/2016	50	9.5.8 execStaticSampling 9.5.9 execDynamicWindowSampling 9.5.7 execEngineeringSampling
2	3	14/11/2016	22	Table 3, Scientific Mode, and the comment under the table
2	3	14/11/2016	55-56	10.1.2 and PAS-ICD-101-001
2	3	14/11/2016	27	PAS-ICD-063-001 modification
2	3	14/11/2016	44	PAS-ICD-093-001 modification
2	3	14/11/2016	31	PAS-ICD-072-001, PAS-ICD-072-002, table 9, 10, remove <accumulationRawDataTime>
2	3	14/11/2016	37-38	PAS-ICD-091-001, table 15 modification
2	3	14/11/2016	47	Table 24, Pulse injection
2	3	14/11/2016	53	9.5.15, Pulse injection
2	3	14/11/2016	45 – 46	Table 23, Pulse injection
2	3	23/11/2016	23	Remove PAS-ICD-081-001, RPW communication
2	3	23/11/2016	68-69	Figure 9 and Normal mode description

2	3	23/11/2016	68-69	PAS-ICD-131-001, PAS-ICD-132-001, Add parameters and commands for CEM calibration
2	3	15/12/2016	33	PAS-ICD-074-001 Format of calibration data is modified
2	3	15/12/2016	62	Tables 28 and 29 changed
2	4	07/04/2017	31	Section 6.5 (Moments calculation) is updated
2	4	07/04/2017	63	Table 28. <staticWindowCalibration> is added, all other fields are modified
2	4	07/04/2017	44	Table 22, values are changed
2	4	07/04/2017	63	Calibration cyclograms is updated
2	4	07/04/2017	59	PAS ON macro is updated
2	4	07/04/2017	58	Table 25 is updated. “Standby” limits.
2	4	07/04/2017	26-27	Table 6 is updated. A note is added.
2	4	21/04/2017	32-33	Table 9 and 10
2	4	21/04/2017	32	PAS-ICD-071-001 : PAS_INTERFERENCE is removed
2	4	21/04/2017	24	Section 5.3.1. Modes coding
2	4	24/04/2017	27	Table 6, MHV limits, CEM HV limits
2	4	24/04/2017	34	Clarification of CALIBRATION format
2	5	26/07/2017	36	PAS-ICD-091-001, Table 14, new constants
2	5	26/07/2017	40	9.2.3, Table 17, Register layout and encoding
2	5	26/07/2017	45	Table 22
2	5	26/07/2017	62	Table 29
2	5	26/07/2017	60	Names of Macro files
2	5	10/09/2017	22	Table 2, HKs layout

2	5	29/08/2017	12	Reference documents, Macros and Cyclogrames
2	5	29/08/2017	65	Cyclogrames files

IV TABLE OF CONTENTS

1. Introduction

1.1. Context of the project

1.1.1. Solar Orbiter mission

The Solar Orbiter is an ESA/NASA mission that will provide an unprecedented opportunity to discover the fundamental connections between the rapidly varying solar atmosphere and the solar wind, revealing the physical links between the outward transport of solar energy, its manifestations in solar convection, the variations of coronal magnetic fields, and the sources and acceleration of solar wind. Solar Orbiter science payload consists of ten instruments including Solar Wind Analyzer suite (SWA).

The objective of SWA is to provide the comprehensive in situ measurements of the solar wind to establish the fundamental physical links between the Sun's highly dynamic magnetized atmosphere and the solar wind in all its quiet and disturbed states. SWA provides high time resolution velocity distributions of solar wind electrons and ions, and ion's composition.

1.1.2. SWA Instruments

SWA suite of instrument is under the responsibility of MSSL and is composed by :

- **HIS (Heavy Ions Sensor) : responsibility of IRAP and SwRI (USA)**
- **PAS (Proton-Alpha Sensor) : responsibility of IRAP**
- EAS (Electron Analyzer System) : responsibility of MSSL
- DPU : responsibility of IFSI

1.1.3. HIS and PAS instruments

PAS

- Will determine the characteristics of the solar wind proton and alpha particle distributions
- Will provide high time resolution solar wind properties like partial 3D and 2D distribution functions, plasma density, plasma bulk velocity and temperature.

HIS

- Will provide critical information on the solar wind composition

IRAP will provide the Electrostatic Analyzer (HIS EAIS) for the HIS sensor, while the "time-of-flight' and detector sections will be provided by US laboratories.

Since HIS EAIS has the same design as the PAS analyzer, HIS EA and PAS instruments will share some documents from the document tree.

1.2. Scope of the document

This document defines the interface between the DPU and PAS instruments.

1.3. General Notes

This document is a supplementary material to the PAS Unit Specification [RD1]. To use this document you have to read firstly the Unit Spec (RD-1) and EAS-PAS FPGA Detailed Design (RD-2).

2. Related documents

2.1. Applicable documents

AD	Reference	Ed./Rev.	Date	Title of the document
AD-1	SOL-EST-RCD-0050-EID-A	5.0	Mar 16 2015	Solar Orbiter EID-A

2.2. Reference documents

RD	Reference	Ed./Rev.	Date	Title of the document
RD-1	SWA-SP-22300-IRAP-059-GEN	3.0	/	SOLAR ORBITER SWA PAS Unit Specification
RD-2	SO-SWA-MSSL-RP-016	/	/	EAS-PAS FPGA Detailed Design Report
RD-3	SO-SWA-MSSL-RQ-010			Solar Orbiter SWA Scientific Operations, Algorithms and Processes Requirements Document
RD-4				PAS_Macro_ON_25_Jul_2017.xlsx
RD-5				PAS_Macro_CONFIGURATION_21_Jun_2017.xlsx
RD-6				PAS_Macro_OFF_21_Jun_2016.xlsx
RD-7				PAS_Macro_DETECTORS_ON_28_Jul_2017.xlsx

RD-8				PAS_Macro_DETECTORS_OFF_21_Jun_2017.xlsx
RD-9				PAS_Macro_Emergency_OFF_21_Jun_2017.xlsx
RD-10				PAS_Normal_mode_Cyclogram_V_2_3_29_Aug_2017.xlsx
RD-11				PAS_Burst_mode_Cyclogram_V_2_0_05_Oct_2016.xlsx
RD-12				PAS_Calibration_cyclogram_06_Apr_2017.xlsx

3. PAS instrument functional overview.

The PAS detailed description is given in RD-1. Here, in this section, we just repeat several very general and simplified points to better understanding PAS functionality. For any real and updated information reader has to refer RD-1.

3.1. Definition of Terms

Sequencer state – the state of the Sequencer that is considered as a State Machine

1/2s tick – a FPGA heartbeat corresponding to a middle way between two full seconds of CSET

Sampling Scheme – instant type of PAS sampling defined by values in the Configuration Table and the running Sequencer routine (Static, Dynamic, Engineering)

Sampling Sequence – sequence of samplings performed during the execution of a specified scheme, preceded by 1s of CEMs HV ramp up and followed by 1s CEMs HV ramp down and 1s idle time interval (see Figure 1).

3D Valid Sampling – sampling that is valid for moments calculation and peak position processing (see Subsection 6.3)

Command – a DPU message to PAS requiring to change PAS configuration or to execute a Scheme. Command name appears in the text as "***execStaticSampling***" for instance.

MAILBOX_FROM_DPU: fixed register in the PAS FPGA shared memory available to DPU write. Used to control Sampling Schemes execution

CONFIGURATION_TABLE: fixed part of the PAS FPGA shared memory where DPU can write. Contains the PAS sequencer configuration values. **This table can be uploaded in the IDLE state of the sequencer only.**

PARAMETER_TABLE: fixed part of the PAS FPGA shared memory available to DPU write. Contains the PAS sampling schemes parameters.

DPU_command – PAS related configuration action of DPU executed by telecommand. It can be a part of Procedure. DPU_command appears in the text as "***setConfiguration***" for example.

Configuration Parameter – parameters needed to PAS control, stored in the DPU memory. It can be updated by DPU_command. Configuration Parameter appears in the text as <transmitHKTimeInterval> for instance.

Mode – a way how DPU creates a final PAS data product by manipulating PAS schemes.

Procedure – a sequence of DPU commands (macro) that make PAS execute some scheme or to manage HVs or do something else. Procedure name appears in the text as: "***PAS_Power_ON***" for example

Cyclogram – a Loop containing execution of one or several procedures. Cyclogram name appears in the text as "**NORMAL mode**" for instance

3.2. PAS sequencer and its schemes

When PAS is ON, its Sequencer code is loaded and running, and DPU has already uploaded PAS's Parameters table and Configuration tables (see RD-1 PAS-R-057-14), PAS can then start to execute some Sampling Scheme (see RD-1 PAS-R-058-*) and produce scientific packets. Sampling Scheme execute on one or several samplings according to specified configuration parameters as **Ne** (number of energy steps), **Nel** (number of elevations), **K** (number of saplings in one second), **N** (length in seconds of the sampling sequence) etc. It is important to say that the PAS Sampling Schemes are NOT PAS Modes with respect to the final scientific data product. PAS Mode (see section 3.3.4 and Section 13 of this document) is Procedure or Cyclogram executed by DPU(see below). For instance to perform the NORMAL (see Sections 11 and 13) mode, DPU has to periodically configurate and start different Sampling Schemes. PAS itself knows nothing about NORMAL mode or BURST mode, and there is no command to PAS like “Start Burst Mode”. The only thing that DPU can do, is to update the corresponding configuration tables and to start the corresponding Scheme at the proper time.

3.3. PAS DPU control principles

3.3.1. PAS run time commands and HV security

PAS real time commands lead to an immediate change of PAS hardware state or modify constants of control circuits. See RD-1 for details. It concerns several static HVs, amplifiers power, operational heating and so on. DPU shall just send the real time commands to PAS. DPU does not check the PAS HV security state like “AirSafe, etc.

3.3.2. PAS Sequencer control commands and sampling basics

These commands are dedicated to update PAS Configuration Tables and Parameter Tables (See RD-1 PAS-R-057-14). These commands can be sent at any time. But PAS process the command only inside 1ms time interval following the **1/s tick**. Execution of such command is possible only when the currently executing sampling scheme is completed.

Generalized diagram of PAS sampling sequence is shown in Figure 1 taken from RD-1.

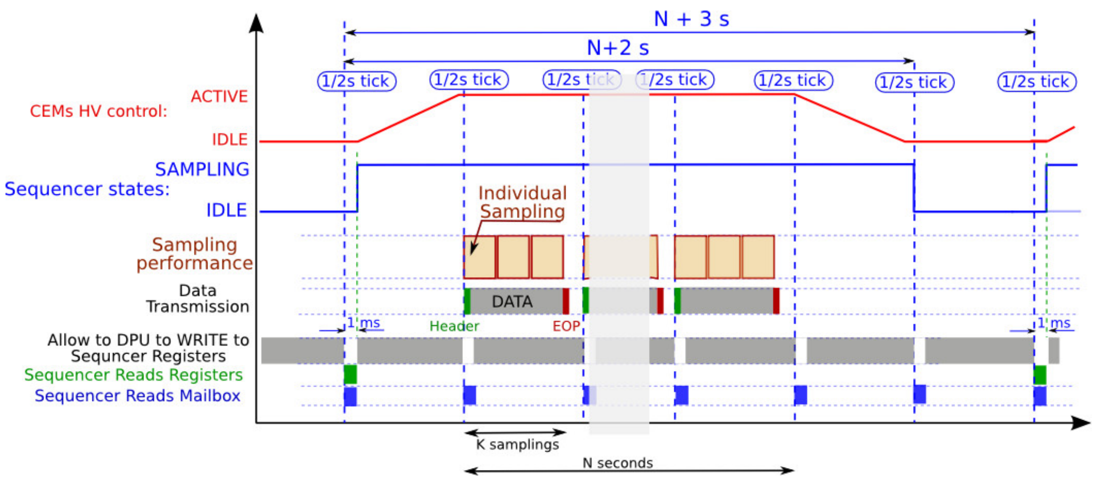


Figure 1 PAS general sampling diagram. Taken from an example of Static Scheme implementation, Figure 32 of RD-1.

DPU can rewrite PAS **PARAMETER_TABLE** at any time but PAS reads the Tables contents and **MAILBOX_FROM_DPU** contents at the green and blue time intervals only. The green intervals indicate the start of new samplings sequence.

- The minimal distance between start one sampling sequence and the next one is 4 sec.
- One scientific packet corresponds to the data accumulated for one second (the real

accumulation time (T(EOP) – T(Header)) can be less than 1s). The maximal packet transmission rate is 1 per sec.

- The maximal packet size transmitted to the DPU is 20770 Bytes. See also subsection 5.3.3.
- One scientific packet can contain data from several samplings

See details in Section 10.

3.3.3. PAS Procedures

A PAS procedure is a sequence of Commands (macros) executed following a specific relative time table. Each procedure has a specific name. PAS does not need procedures with arguments. The List of the Procedures is in section 11. PAS procedures consists of PAS or DPU commands with arguments that are "Configuration Parameters". All "Configuration Parameters" can be updated by a DPU procedure or telecommand. Later in the test the commands dedicated to update configuration parameters are identified as an assignment operation: "<configurationParameter> = {}".

3.3.4. PAS Modes and Cyclograms

Generally speaking there are two PAS scientific modes: **NORMAL** and **BURST** (both with some options). Both modes are the result of the execution by DPU of a dedicated **Cyclogram** that is a Procedure containing a loop of PAS commands. If no PAS scientific mode is running, PAS FPGA is in IDLE state and generates only HK packets. To check PAS fuctionality we have an ENGENEERING mode (see section 11) and also perform a special CALIBRATION macro.

The NORMAL mode is a 4s period cadence of Static Scheme samplings with the insert of short Dynamic Scheme intervals (calling "Snapshot") every 300s. Also every 100s (except time of Snapshot) one or several Full 3D sampling are executed. See a simple diagram of NORMAL mode idea in Figure 2. See the detail modes and cyclograms description in Section 12.

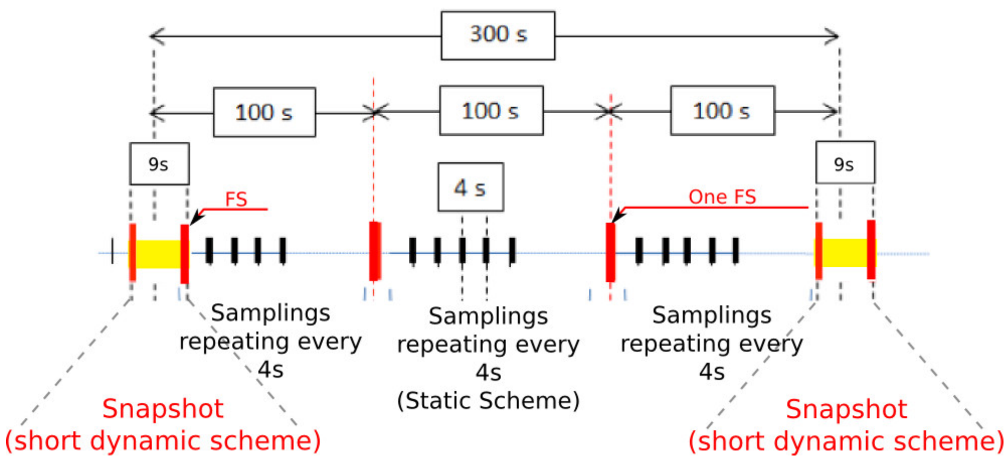


Figure 2 : NORMAL mode generalized diagram. Vertical red bars show one or several FullSampling

A BURST mode is an execution of continuous samplings (with no gaps in the data flow) in Dynamic Window or Static Schemes.

It is important to mention here that in Static Scheme DPU has to calculate Sampling window position (see Section 4.2 of PAS Unit Spec) from the latest FullSampling (see Section 6.4).

The ENGENEERING mode allows to execute any PAS related commands, including PAS procedures and forward to the ground any scientific and housekeeping packets with or without compression.

4. PAS States

4.1. PAS FPGA states description

A detailed description of FPGA states is presented in the Unit Spec (RD-1, **PAS-R-058-001, Table 34**). It is important to say that when the Sequencer initialization is completed, the Sequencer can be in only two states: IDLE, and SAMPLING. Transfer from IDLE state to SAMPLING state is controlled by mailbox content (see Table 34 of RD-1). The Sequencer can read the updated parameters and configuration tables only if it is in the IDLE mode. The mailbox contents is checked every second (at the end of **1/2s** tick, see Figure 1) but only 0xFF (***abortExecution***) leads to immediate stop of the sampling and transfer the Sequencer to the IDLE state. The Sequencer takes into account other mailbox values in the IDLE state only.

4.2. States transition mechanism

To control the sequencer execution, DPU must use a dedicated mailbox in the shared memory (address 0x7E). The access to the mailbox is implemented by following commands:

- ***abortExecution,***
- ***execEngineeringSampling,***
- ***execStaticSampling,***
- ***execDynamicWindowSampling***
- ***gotoIdleMode.***

These commands are described in Section 11. The sequencer reads the mailbox exactly at every **1/2s**. If DPU rewrite the mailbox during next 1ms after the **1/2s** heartbeat, Sequencer ignores mailbox contents and waits next **1/2s** tick to read the mailbox. See section 10.2.

When the sequencer is in IDLE mode and the Mailbox indicates to start sampling, the Sequencer switches to sampling mode immediately after reading the mailbox content. Then the sequencer executes the requested scheme using the corresponding parameters table. The sequencer comes back automatically to the IDLE mode at the end of the scheme execution. If the mailbox has not been changed during the previous run, the sequencer enters again in sampling mode and executes again the previous scheme.

To quit the sampling mode, DPU can send the ***abortExecution*** command to abort immediately scheme execution (at the end of the current **1/2s** time interval) or send a ***gotoIdleMode*** command to stay in IDLE mode when the scheme execution is finished.

4.3. Time synchronization

All the Sequencer activity is synchronized with **1/2s** tick. See RD-1 **PAS-R-057-012** and Figures 31 – 34.

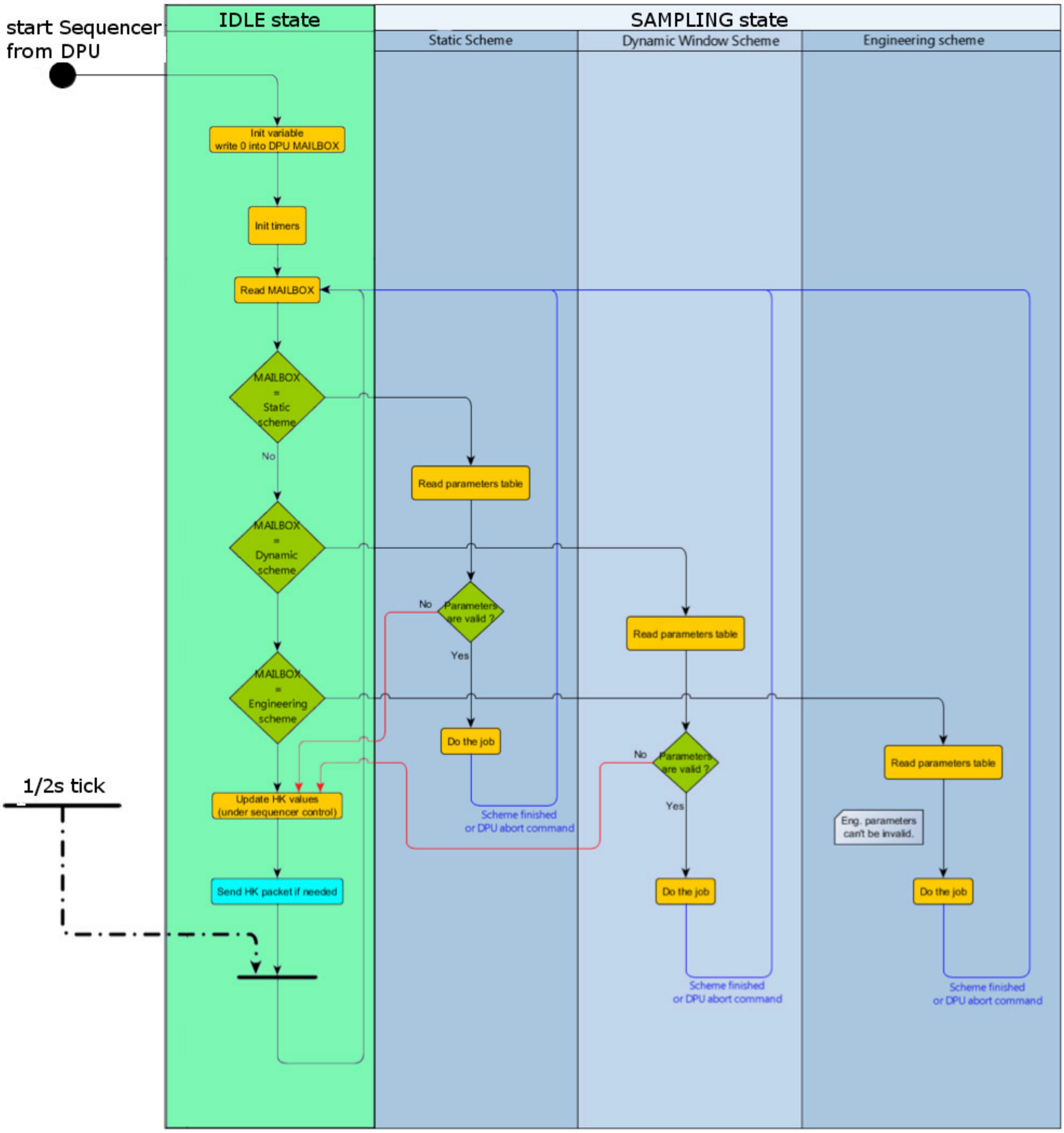


Figure 3 : Sequencer states and schemes diagram

5. Transactions management

5.1. Commands

Command format is described in RD-2. Here is just a reference information. The command format is shown in Table 1. Actually only four type of commands are available:

- READ shared memory field
- WRITE shared memory filed
- ENABLE (for HV operations)
- GO (for HV operations)

The corresponding addresses of the shared memory are described in Section 9.

Byte	Length, bytes	Value	Description
1	1	variable	Command type: 0x00 – ENABLE 0xC0 – GO 0x80 – WRITE 0xA0 – READ
2 - 3	2	variable	ADDRESS if need
4 - 5	2	variable	Data length N, in bytes for WRITE, N = 3 (not mandatory) for READ
6 – 6+N-1	N		Data, 24bits words, MSB first for WRITE, or 3 bytes of reply length, in bytes
6+N	1		EOP

Table 1. DPU command layout

Specific requirements for DPU are as follows:

PAS-ICD-051-001:

1. DPU shall terminate each command with End Of Packet (EOP).
2. All commands generate a response: ACK, NACK, or requested data (for "***getSomething***" commands). The descriptions of ACK, NACK or requested data responses are in RD-2.
3. When DPU is going to perform WRITE to the FPGA sequencer dedicated tables, it shall firstly write 0x00 to **MAILBOX_FROM_DPU** byte (see Section 9), then to perform the main WRITE, and then to write 0x0N (Scheme number) to the **MAILBOX_FROM_DPU** to resume the sampling. This mechanism allows to protect the sequencer tables. **IMPORTANT EXCEPTON**: see the requirement PAS-ICD-095-001 (section 9.5.11) about WRITE to the Stepping Configuration Table. This command can be executed only when PAS FPGA is in the IDLE state. Thus DPU shall read the PAS state and wait the IDLE before to perform the WRITE.
4. DPU shall do not execute WRITE to the Sequencer dedicated registers during the first 1ms immediately after **1/2s** heartbeat.

5.2. HK packet manipulation

5.2.1. Packet description

HK packet layout is as follows (Table 2):

Byte(s)	Bites	Value	Description	DPU readable
1		0x02	PAS ID	-
2		0x42	PAS HK packet ID	-
3,4		0xFF80	Packet type	-
5, 6		0x0053	Data field length in bytes	-
7(MSB)-12			Time stamp 48 bits (see PAS-R-057-012)	-
13(MSB)-14		Variable	V-MON-C (Central CEM HV)	YES
15(MSB)-16		Variable	V-MON-L (Periphery CEM HV)	YES
17(MSB)-18		Variable	I-MON-C (Central CEM current)	YES
19(MSB)-20		Variable	I-MON-L (Periphery CEM current)	YES
21(MSB)-22		Variable	T-Mon-C (Central CEM temperature)	YES
23(MSB)-24		Variable	T-Mon-L (Periphery CEM temperature)	YES
25(MSB)-26		Variable	T1_HEATER	-
27(MSB)-28		Variable	T2_HEATER	-
29(MSB)-30		Variable	+24V_CEM_OUT	YES
31(MSB)-32		Variable	+5V_CEM_OUT	YES
33(MSB)-34		Variable	+12V_HT_OUT	YES
35(MSB)-36		Variable	-12V_HT_OUT	YES
37(MSB)-38		Variable	+3V3_FPGA_OUT (FPGA + CEM)	YES
39(MSB)-40		Variable	1V5_FPGA_OUT	YES
41(MSB)-42		Variable	TEMP_DCDC	YES
43(MSB)-		Variable	TEMP_FPGA	YES

44				
45(MSB)-46		Variable	HK_I_+24V_CEM	YES
47(MSB)-48		Variable	HK_I_+5V_CEM	YES
49(MSB)-50		Variable	HK_I_+12V_HT (DC-DC current, HVPS and CEM)	YES
51(MSB)-52		Variable	HK_I_-12V_HT (DC-DC current, HVPS and CEM)	YES
53(MSB)-54		Variable	HK_I_3V3_FPGA (FPGA + CEM)	YES
55(MSB)-56		Variable	HK_I_+28V_PRI (DC-DC input current)	YES
57(MSB)-58		Variable	HK_I_1V5_FPGA	YES
59(MSB)-60		Variable	T3_HEATER	
61(MSB)-62		Variable	TEMP_HVPS	YES
63(MSB)-64		Variable	TEMP_EA	YES
65(MSB)-66		Variable	HK_MHV_POS	-
67(MSB)-68		Variable	MHV_NEG	-
69(MSB)-70		Variable	HK_ANL_HK	-
71(MSB)-72		Variable	HK_TOP_DEFL	-
73(MSB)-74		Variable	HK_TOP_CAP	YES
75(MSB)-76		Variable	HK_BOT_DEFL	-
77(MSB)-78	15		Heater HK channel Select	-
	14		OP_HEATER_ON	-
	13		1 – Sequencer in state RUNNING	-
	12		1 – UPLOADED, see section 5.8	-
	11		Pre-amp1 over current	YES
	10		Pre-amp 2 over current	YES
	9		HV_Disable	-
	8		HV_Airsafe	-

	7 – 0		Memory_error_counter	-
79(MSB)-81	23		IDLE1	-
	22		IDLE2	-
	21		ANALYSER_GAIN	-
	20		TOP_CAP_GAIN	-
	19		TOP_DEF_GAIN	-
	18		BOTTOM_DEF_GAIN	-
	17		TOP_CAP_SIGN	-
	16		TOP_DEF_SIGN	-
	15		BOTTOM_DEF_SIGN	-
	14		Analyzer validity flag	-
	13		Bottom deflector validity flag	-
	12		Top deflector validity flag	-
	11		Top cup validity flag	-
	4 - 10		Energy step (0 – 95)	-
	0 - 3		Elevation bin (0 – 8)	-
82(MSB)-88			Reserved	-
89		0x100	EOP	

Table 2 HK packet contents and layout

- Blue parameters are read by PAS FPGA with 1s sampling interval
- Red parameters are read by the Sequencer command
- Black parameters are read with 32 s sampling interval

5.2.2. Analyzer, top/bottom deflectors and top cup High Voltages decoding

Attention: This section is for reference only. DPU has no duty to decode these HV values and check them. These values are sweeping very fast and they are subject of ground processing.

To decode the real values of HV applied to Analyser, Top Deflector, Botom Deflector and Top Cup of the PAS sensor the following calculations shall be made:

"Analyzer HV HK" = HK_ANL * (32 if ANALYSER_GAIN == 1)
"Top Deflector HV HK" = HK_TOP_DEFL * (-1 if TOP_DEF_SIGN == 1) * (32 if TOP_DEF_GAIN == 1)
"Bottom Deflector HV HK" = HK_BOT_DEFL * (-1 if BOTTOM_DEF_SIGN == 1) * (32 if BOTTOM_DEF_GAIN == 1)
"Top Cup HV HK" = HK_TOP_CAP * (-1 if TOP_CAP_SIGN == 1) * (32 if TOP_CAP_GAIN == 1)

Such decoding makes sense **if the corresponding validity flag == 1**. If the validity flag is set to 0 it means the SIGN and GAIN flags are not synchronized with ADC output.
Note that each new HK packet contains a new HV HK set corresponding to successively increasing energy step and elevation bin numbers.

5.3. Scientific Packet manipulation

PAS forms one packet during one sec. One packet can contain several samples and variable number of sub packets. Full description of the Scientific packet generation is described in **PAS-R-057-010** and in **PAS-R-058-002, PAS-R-058-003, PAS-R-058-005**.
The global structure of one second data transaction is as follows:

- Scientific Data Header that contains number expected subpackets
- Science Packet Leader contains the real data size in bytes
- Subpacket
- Subpacket
- …………..
- Subpacket
- EOP

5.3.1. Scientific Data Header Layout

Byte(s)	Bites	Value	Description
1		0x02	PAS ID
2		0x42	PAS science packet ID
3,4		0xFF00	Packet ID
5,6		0x000E	Fixed length of Science header
7	0 – 5	variable	K value
8		variable	Rotating count
9 - 14		variable	Time stamp 48 bits (see PAS-R-057-012)
15(MSBs) - 17	23-17	Variable	FIRST_ENERGY, Sequencer register
	16-10	Variable	ENERGY_NUMBER, Sequencer register
	9-6	Variable	FIRST_ELEVATION, Sequencer register
	5-2	Variable	ELEVATION_NUMBER, Sequencer register
	1	Variable	CEM_GLAG 0 – all CEMs enabled, 1- central CEMs enabled
	0	Variable	1 - Enable FPGA to calculate Max Count Position
18(MSBs) - 20	23	0	
	22 -17	Variable	MAX_CNT_ENERGY (last value taken into account by Dynamic Scheme)
	16	0	

	15 - 12	Variable	SCIENTIFIC MODE (index provided by DPU)*
	11 – 8	Variable	MAX_CNT_ELEVATION
	7 - 6	0	
	5	Variable	0 – Static Scheme, 1 – Dynamic Scheme
	4	Variable	1 – Full3D during Dynamic Scheme, 0 – fast part of Dynamic Scheme, void during Static Scheme
	3 - 0		MAX_CNT_CEM
21		0x100	EOP

Table 3: Science Data Header Layout

* The SCIENTIFIC MODE index is bit-to-bit image from bits 12…15 of the commands execXXXSampling. See 9.5.7, 9.5.8, and 9.5.9. The contents is defined by DPU. The values (for reference only) are as follows:

- 0 – NA for PAS
- 1 – Normal Mode
- 2 – NA for PAS
- 3 – Burst Mode
- 4 – Engineering Mode
- 5 - Calibration Mode

5.3.2. Science Data Subpacket layout

The first transmitting subpacket is the "Science Packet Leader" (Table 4). Then, after completing of each elevation step, PAS dumps the Science Subpacket (Table 5). Number of Subpackets = ENERGY_NUMBER × ELEVATION_NUMBER × K is variable. This cadence is followed by EOP.

Byte(s)	Bites	Value	Description
1		0x02	PAS ID
2		0x42	PAS science packet ID
3,4		0xFF10	Packet ID
5,6		Variable	Data field length in Bytes = ENERGY_NUMBER × ELEVATION_NUMBER × K × 24
7	0 – 5	variable	K value
8		variable	Rotating count
9 - 11		0x0	

Table 4 Science Packet Leader layout

Byte(s)	Bites	Value	Description
0(MSB), 1	15	0	
	14 – 8	Variable	ENERGY_BIN

	7 – 4	Variable	ELEVATION_BIN
	3 – 1	000	
	0	Variable	0 – all CEMs enabled, 1- central CEMs enabled
2 - 3		Variable	CEM 0 count
4 - 5		Variable	CEM 1 count
…….			………..
22 - 23			CEM 10 count

Table 5 Scientific Subpacket layout

5.3.3. PAS data production rate

To calculate the average PAS sensor data production rate, the following formula shall be used:

PAS rate [Bytes/s] = N * (33 + (24*Ne*Nel)*K) / (N + 3)

The maximal packet size transmitted to the DPU is 20770 Bytes

6. Requirements for DPU data processing

6.1. Readable HK parameters

PAS-ICD-061-001:

DPU shall be capable to decode PAS HK packed runtime for FDIR purposes. The readable parameters are indicated in Table 2 (right column). The corresponding default limits are shown in Table 6. The limits shall be stored as DPU Configuration Parameters updatable by telecommand.

DPU shall provide a possibility to update any of Configuration Parameters in Table 6 by telecommands.

Parameter Name	HK limits variables with default values
V-MON-C	<minCemHV> = 700 <maxCemHV> = 1500
V-MON-L	<minCemHV> = 700 <maxCemHV> = 1500
I-MON-C	<maxCEMcI> = 2000 (250uA)
I-MON-L	<maxCEMpI> = 700 (85uA)
T-Mon-C, T-Mon-L	<maxTCEM> = 2714 (50°C)
+24V_CEM_OUT	<minCEM24V> = 3003 (22V)
+5V_CEM_OUT	<minCEM5V> = 2972 (4.5V)
+12V_HT_OUT	<minHVp12V> = 3127 (11.5V) <maxHVp12V> = 3725 (13.7V)
-12V_HT_OUT	<minHVm12V> = 3725 (-13.7V) <maxHVm12V> = 3127 (-11.5V)
+3V3_FPGA_OUT	<minFPGA3V3> = 2621 (3.2V) <max FPGA3V3> = 2785 (3.4V)
1V5_FPGA_OUT	<minFPGA1V5> = 1204 (1.47V) <max FPGA1V5> = 1278 (1.56V)
TEMP_DCDC	<maxTdcdc> = 3243 (70°C)
TEMP_FPGA	<maxTfpga> = 3243 (70°C)
HK_I_+24V_CEM	<maxCEM24curr> = 815 (40mA)
HK_I_+5V_CEM	<maxCEM05curr> = 2162 (60mA)

HK_I_+12V_HT	<maxHVp12curr> = 3976 (95mA)
HK_I_-12V_HT	<maxHVm12curr> = 3276 (80mA)
HK_I_+3V3_FPGA	<maxFPGAp12curr> = 815 (40mA)
HK_I_+28V_PRI	<max28curr> = 1753 (240mA)
HK_I_1V5_FPGA	<maxFPGA2p5curr> = 2463 (210mA)
HK_MHV_POS	<minMainHVpos> = 4033 <maxMainHVpos> = 4095
HK_MHV_NEG	<minMainHVneg> = 4033 <maxMainHVneg> = 4095
TEMP_HVPS	<maxEATemp> = 2547 (65°C)
Pre-amp1 over current	0 (Not Active)
Pre-amp 2 over current	0 (Not Active)

Table 6. Limits of the readable HK parameters. Right column contains names of corresponding Configuration Parameters.

Notes:

1. Limits are expressed in the decimal values

2. PAS HK system can measure either full <cemHV> value either jest ½ of this value. The last one is also correct. DPU shall recognize the values falling out of these two bands as invalid.

6.2. Internal DPU FDIR actions

PAS-ICD-062-001:

Internal DPU actions in case of HK readable parameters inconsistent shall be as shown in Table 7.

DPU shall provide possibility to inhibit any action described in Table 7 by DPU command (inside the procedure) or by telecommand.

DPU FDIR Action and Its reason	What to do in case of out of limits	Name of Procedure
absenceHK Absence of the HK inside the 11.5s interval while Sequencer is running	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
ampOvercurrent Pre-amp 1 over current Pre-amp 2 over current	OFF and then ON again the Detectors Indicate the reason in the error message	***PAS_DETECTORS_OFF*** ***PAS_DETECTORS_ON***

cemOvercurrent HK_I_+24V_CEM	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
hvOvercurrent HK_I_+12V_HT	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
hvpsOvercurrent HK_I_-12V_HT	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
pasOvercurrent HK_I_+3V3_FPGA HK_I_1V5_FPGA HK_I_+28V_PRI	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
hvPowerFail +24V_CEM_OUT +5V_CEM_OUT +12V_HT_OUT +12V_HT_OUT	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
cemHVfail V-MON-C V-MON-L I-MON-C I-MON-L DPU shall register ONLY maximal value over 300s time interval	OFF routinly Indicate the reason in the error message	***PAS_DETECTORS_OFF*** ***PAS _OFF***
mainHVfail HK_MHV_POS HK_MHV_NEG	OFF immediately Indicate the reason in the error message	***PAS_EMERGENCY_OFF***
cemOverHeat T-Mon-C, T-Mon-L greater than <maxTCEM>	OFF routinly Indicate the reason in the error message	***PAS_DETECTORS_OFF*** ***PAS _OFF***
pasTempOutLim TEMP_DCDC TEMP_FPGA TEMP_HVPS	OFF routinly Indicate the reason in the error message	***PAS_DETECTORS_OFF*** ***PAS _OFF***

Table 7

6.3. Scientific data processing general requirements

Definition of "3D Valid Sampling":

The properties of "3D Valid Sampling" are as follows:

- Ne >= <Ne_valid> = 48 by default
- Nel >= <Nel_valid> = 5 by default

These values shall be updatable in flight.

Configuration parameters related to the Data Processing:

Configuration Parameter	Description	Default Value
<cemMask>	See 6.4	{1,1,1,1,1,1,1,1,1,1,1}

Table 8

PAS-ICD-063-001:

1. DPU shall provide possibility to update Configuration Parameter shown in Table 8 by dedicated telecommand.
2. DPU shall calculate moments for "3D Valid Samplings" only.
3. DPU shall calculate the peak position for "3D Valid Samplings" only.
4. DPU shall be able to compress any type of sampling.

6.4. Peak position determination and new sampling window position determination

PAS-ICD-064-001:

DPU shall calculate the PAS count maximum position from each FULL 3D Sampling (see RD-1) and keep this value until next FULL 3D Sampling arriving. This value shall be used to calculate **Se** and **Sel** and update <staticWindow> (see Table 23) before to run StaticScheme. Note, that according to PAS-ICD-093-001 <staticWindow> can be updated by telecommand.

The variable : <maxCountPosition>

Content: int[4] = { Max_Cnt_Ener, Max_Cnt_Elev, Max_Cnt_CEM, Max_Cnt}

Ranges: [0 – 95, 0 – 8, 0 – 10, 0 – 2^16-1]

Init Values: [64,4,5,0]

Description: [Energy bin corresponding to the maximal ion flux,
Elevation bin corresponding to the maximal ion flux,
CEM number corresponding to the maximal ion flux,
Maximal Count]

The Configuration Parameter: <cemMask>

Content: int[11]

Default: [1,1,1,1,1,1,1,1,1,1,1]

Description: Mask to obscure some CEMs if need.

The <maxCountPosition> shall be calculated as follows:

```
short CounArr[96][9][11];     // [Energies][Elevations][CEMs]
short cemMask[11];            // Configuration parameter, 1 – inhibit the
channel
short Max_Cnt_Ener = 0; Max_Cnt_Elev = 0; Max_Cnt_CEM = 0; Max_Cnt ;
short se, sel, ne,nel;        // start points of sampling window, window
dimentions
short ie, iel, icem;
short OldCount = 0;

for(ie = se; ie < se+ne; ie++)
{
  for(iel = sel; iel < sel+nel; iel++)
  {
     for(icem = 0; icem < 11; icem++)
       if(OldCount < CountArr[ie][iel][icem]*(cemMask[icem] == 0);
       {
            OldCount = CountArr[ie][iel][icem];
            Max_Cnt_Ener = ie;
            Max_Cnt_Elev = iel;
            Max_Cnt_CEM = icem;
            Max_Cnt = OldCount ;
       }
  }
}
```

The values Se and Sel for new window position shall be calculated as follows:

Sel = Max_Count_Elev – Nel/2 (>= 0)

Se = Max_Count_Ener – Ne*0.61 (>= 0)

The illustration of the sampling window position is in Figure 4.

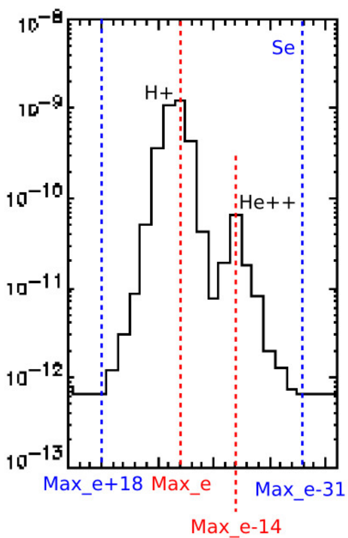


Figure 4 Solar wind spectrum example and corresponding sampling window position

6.5. Ion Moments calculation

Ion moments determination algorithm is described in RD-3. The details of the moment calculation algorithm and corresponding constants are described in RD-13.

6.6. Data Compression

Data compression requirements are given in RD-3

7. Telemetry packets generation and format requirements

7.1. Data Products

PAS-ICD-071-001:

DPU shall produce:

PAS_DUMP – PAS FPGA memory dump generated exclusively after execution of the command ***shareMemoryDump*** or ***getSequencerDebugValue*** and using PID 99 memory dump service

PAS_HK – PAS HK packet containing the last HK information received from PAS. This packet shall contain all information from the original PAS HK. This packet shall be forwarded to the SO DMS by three alternative ways

1. Immediate forward as soon as DPU receives the corresponding HK packet from PAS, decode and process it.
2. Immediate forwarding of PAS HK received after ***hkRequest*** command.

PAS_RAW_UC – PAS raw uncompressed data containing the last data received from PAS. The maximal quant of data accumulation is one second.

PAS_RAW_C – PAS raw compresses data containing the last data received from PAS. The maximal quant of data accumulation is one second.

PAS_MOM – PAS ion moments as described in RP-3.

PAS_CALIBR – Low Latency CEM calibration data

7.2. Raw data telemetry formats

PAS-ICD-072-001:

The uncompressed raw data universal format shall contain at least information as shown in Table 9. This format shall be used for **PAS_RAW_UC** and **PAS_ACC_UC** data products

Type	Length (bytes)	Description
Fixed length	2	SSID Counter
Header	6	Time Stamp (== to Time stamp of the Original PAS packet)
	3	FIRST_ENERGY, ENERGY_NUMBER, FIRST_ELEVATION, ELEVATION_NUMBER, CEM_FLAG, SAMPLING_SCHEME
	1	K
	1	MAX_CNT_ENERGY
	1	MAX_CNT_ELEVATION

	1	MAX_CNT_CEM
Variable length Data	2 TBD Variable	SSID Counter N bytes to follow in this packet Raw data PAS_Counts[ENERGY_NUMBER,ELEVATION_NUMBER,11,K] (Contents of the Table 5 without auxiliary data)

Table 9

PAS-ICD-072-002:

The compressed raw data universal format shall contain at least information as shown in Table 10. This format shall be used for **PAS_RAW_C** and **PAS_ACC_C** data products

Type	Length (bytes)	Description
Fixed length Header	2 6 3 1 1 1 1	SSID Counter Time Stamp (== to Time stamp of the Original PAS packet) FIRST_ENERGY, ENERGY_NUMBER, FIRST_ELEVATION, ELEVATION_NUMBER, CEM_FLAG SAMPLING_SCHEME K MAX_CNT_ENERGY MAX_CNT_ELEVATION MAX_CNT_CEM
Variable length Data	2 TBD Variable	SSID Counter N bytes to follow in this packet Compressed PAS_Counts[ENERGY_NUMBER,ELEVATION_NUMBER,11,K] (Contents of the Table 5 without auxiliary data)

Table 10

7.3. Ion moments data telemetry format

PAS-ICD-073-001:

PAS moments data format shall contains at least fields as shown in Table 11

Type	Length (bytes)	Description
Fixed length Header	2 6 3 1	SSID Counter Time Stamp (== to Time stamp of the Original PAS packet) FIRST_ENERGY, ENERGY_NUMBER, FIRST_ELEVATION, ELEVATION_NUMBER, CEM_FLAG SAMPLING_SCHEME K
Fixed length Data		25 following structures (All values are 16 bits integers) :

	2
	2
	2
	2
	2
	2
	2
	2
	2
	2
		{
		N – Number Density
		N_VX – Velocity
		N_VY – Velocity
		N_VZ – Velocity
		N_P_XX – Pressure tensor
		N_P_YY – Pressure tensor
		N_P_ZZ – Pressure tensor
		N_P_XY – Pressure tensor
		N_P_XZ – Pressure tensor
		N_P_YZ – Pressure tensor
		}

Table 11

7.4. PAS Low Latency Calibration Data Format

PAS-ICD-074-001:

PAS CEM Calibration Data shall be transferred in one TM packet contain at least data as shown in Table 12. N is the number of CEM HV points calculated as N = (<CEMcalibrHVStop> - (<CEMcalibrHVStart>)/<CEMcalibrHVStep> (see Table 28 in page 60).

Type	Length (bytes)	Description

Fixed length data	2	SSID Counter
		The N+1 blocks containing:

	6	Time Stamp (== to Time stamp of the Original PAS packet)
	1	<Sample_Number> (see Table 28)
	8	<maxCountPosition> for <CEMcalibr_mask_0> (4 of 16 bits integers) (see Table 28)
	8	<maxCountPosition> for <CEMcalibr_mask_1> (4 of 16 bits integers)
	8	<maxCountPosition> for <CEMcalibr_mask_2> (4 of 16 bits integers)
	8	<maxCountPosition> for <CEMcalibr_mask_3> (4 of 16 bits integers)
	8	<maxCountPosition> for <CEMcalibr_mask_4> (4 of 16 bits integers)

Table 12

8. PAS service 20 communication

8.1. PAS solar wind parameters distribution

PAS-ICD-081-001:

PAS has to provide to Service 20 solar wind moments values as shown in Table 13

Data length, bytes	Data Description
	All Values are 16 bits integer
2	N – density
2	N_VX – Velocity
2	N_VY
2	N_VZ

Table 13

9. PAS Commands description

9.1. List of the Configuration Parameters related to the PAS Configuration commands.

PAS-ICD-091-001:

DPU shall provide possibility to update the Configuration Parameters shown in Table 14 used as arguments in the native PAS configuration commands (section 9.2) by dedicated telecommands and/or by macrp script..

The following table (not numbered) contains the named constants than can be used in macro scripts to define the Configuration Parameters assignment:

Constant Name	Value	Description
<ampPowerONC>	0x000001 bit 0 Pre_amp1_pwr_off= 1	Constant to give a value of <amplPower> configuration parameter Center configuration
<ampPowerONF>	0x000003 bit 0 Pre_amp1_pwr_off= 1 bit 1 Pre_amp2_pwr_off= 1	Constant to give a value of <amplPower> configuration parameter Full configuration
<ampPowerOFF>	0x000000 bit 0 Pre_amp1_pwr_off= 0 bit 1 Pre_amp2_pwr_off= 0	Constant to give a value of <amplPower> configuration parameter Idle configuration.
<hvSafetyActive>	**0x000007** bit 0 HV-DISABLE = 1 (CEMs ON) bit 1 HV_Airsafe = 1 (FULL HV) bit 2 ON_HV_ea_fpga = 1 (ON HVPS) bit 3 HV_DIS_1 = 0 (centr. CEMs ON) bit 4 HV_DIS_2 = 0 (peref. CEMs ON) bit 8 HEATER ON = 0 bit 14 HEATER_LOOP_ON = 0 (OFF)	Constant to give a value to <hvSafety> configuration parameter. Active flight configuration.
<hvSafetyActiveGND>	**0x000005** bit 0 HV-DISABLE = 1 (CEMs ON) bit 1 HV_Airsafe = 0 (10% of HV) bit 2 ON_HV_ea_fpga = 1 (ON HVPS) bit 3 HV_DIS_1 = 0 (centr. CEMs	Constant to give a value to <hvSafety> configuration parameter. Ground test configuration.

	ON) bit 4 HV_DIS_2 = 0 (peref. CEMs ON) bit 8 HEATER ON = 0 bit 14 HEATER_LOOP_ON = 0 (OFF	
<hvSafetyOFF>	**0x00001A** bit 0 HV-DISABLE = 0 (CEMs OFF) bit 1 HV_Airsafe = 1 (FULL HV) bit 2 ON_HV_ea_fpga = 0 (OFF HVPS) bit 3 HV_DIS_1 = 1 (centr. CEMs OFF) bit 4 HV_DIS_2 = 1 (peref. CEMs OFF) bit 8 HEATER ON = 0 bit 14 HEATER_LOOP_ON = 0 (OFF)	Constant to give a value to <hvSafety> configuration parameter. Initial (OFF) configuration
<hvSafetyOFFGnd>	**0x000018** bit 0 HV-DISABLE = 0 (CEMs OFF) bit 1 HV_Airsafe = 0 (10% of HV) bit 2 ON_HV_ea_fpga = 0 (OFF HVPS) bit 3 HV_DIS_1 = 1 (centr. CEMs OFF) bit 4 HV_DIS_2 = 1 (peref. CEMs OFF) bit 8 HEATER ON = 0 bit 14 HEATER_LOOP_ON = 0 (OFF)	Constant to give a value to <hvSafety> configuration parameter. Initial (OFF) configuration

Configuration parameters table

Parameter Name	Default Value	Description, Comments
<amplPower>	<ampPowerOFF>	Argument of the command ***setPreAmpPowerRegister*** See Table 16 and 9.2.2
<hvSafety>	<hvSafetyOFF>	Argument of the command ***setMainControlRegister*** See Table 16 and 9.2.3
<pidCoef>	0x000000	Argument for the command ***setRegulationCoeff***. See table 16 and 9.2.4
<openLoopDuty>	0x001000 bits 0-11 Duty cycle = 0	Argument for the command ***setOpenLoopDuty***. See Table 16 and 9.2.4.

	bit 12 HEATER_OFF = 1 (OFF)	
<heaterSetup>	0x005000 bits 0 – 11 Target temperature bits 12-18 Heather Sensor # = 5	Argument for the command ***setRegulatioSetPoint***. See Table 16 and 9.2.4.
<cemMask>	0x000000	Argument for the command ***setPeakTrackingMask***. See Table 16 and 9.2.6
<cemHV>	0x000000	Argument of the command. ***xxSetHV_CEM*** See Table 16 and 9.2.7
<mainHV>	0x000000	Argument of the command. ***xxSetMAIN_HV*** See Table 16 and 9.2.8
<sequencerFile>		Code file of 0x1800 size

Table 14

9.2. PAS Configuration commands

Configuration Command is a native command that DPU sends to PAS to control HV, power, heaters, etc. These commands control all PAS functionality that are running independently from the Sequencer activity.

9.2.1. Commands list

PAS-ICD-092-001:
DPU shall provide Configuration Commands as shown in Table 15.

Commands	Argument	Description	Conditions	Return
setPreAmpPowerRegister	<amplPower>	Set preamplifiers power to ON/OFF	~	ACK/NACK
getPreAmpPowerRegister		Get the current preamplifiers power states	~	Data requested
setMainControlRegister	<hvSafety>	Configure PAS HV enable/disable and air safe protection	~	ACK/NACK
getMainControlRegister		Get the current main control register value	~	Data requested
setRegulationCoeff	<pidCoef>	Configure the PID Kp and Ki coefficients for the operational heater control loop	~	ACK/NACK
getRegulationCoeff		Get the current PID coefficients	~	Data requested

setRegulationSetPoint	<heaterSetup>	Set the Operational Heater target temperature	~	ACK/NACK
getRegulationSetPoint		Get the current regulation setPoint parameters	~	Data requested
setOpenLoopDuty	<openLoopDuty>	Set duty cycle value of Open Loop heater control (addr 0x5000)	~	ACK/NACK
getOpenLoopDuty		Get the current duty cycle	~	Data requested
hkRequest		Request an HK packet	Sequencer is not running	ACK+ HK packet
setPeakTrackingMask	<cemMask>	Configure the peak tracking mask	~	ACK/NACK
getPeakTrackingMask		Get the current peak tracking mask	~	Data requested
enableSetHV_CEM ***goSetHV_CEM***	<cemHV>	Configure the HV_CEM with a pair of command	Sequencer must be running	ACK/NACK
enableSetMAIN_HV ***goSetMAIN_HV***	<mainHV>	Configure the MAIN_HV with a pair of command	Sequencer must be running	ACK/NACK
clearSharedMemory		Fill all shared memory to 0. Must be done before starting a sequencer.	~	ACK/NACK
uploadSequencer	<sequencerFile>	Upload the sequencer code	Sequencer is not running	ACK/NACK
getSequencer		Get back the sequencer code	Sequencer is not running	Data requested
startSequencer		After a sequence table has been uploaded the FPGA will allow the DPU to send a 'start' command to start the sequencer.	Sequencer is not running	ACK/NACK

Table 15

9.2.2. ***PreAmpPowerRegister*** commands

The ***setPreAmpPowerRegister*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2801 16 bits		size=0x0003 16 bits		<amplPower> 24 bits		

Argument = <amplPower> as follows :

Bit	<amplPower> desctiption	Bit set to 0	Bit set to 1
0	Pre_amp1_pwr_off (OFF centrals CEMs amplifiers)	OFF	ON
1	Pre_amp2_pwr_off (OFF periphery CEMs amplifiers)	OFF	ON
2-23	Not used	~	~

Table 16

The ***getPreAmpPowerRegister*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x2801 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the current OnOffs value.

9.2.3. *MainControlRegister* commands

The ***setMainControlRegister*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2800 16 bits		size=0x0003 16 bits		<hvSafety> 24 bits		

Argument = <hvSafety> is as follows

Bit	Descriotion	Bit set to 0	Bit set to 1
0	HV_DISABLE (all CEMs HV OFF)	CEMs OFF	CEMs ON
1	HV_Airsafe (CEM and MainHV 1/10 of target)	10% of HV specified	FULL HV
2	ON_HV_ea_fpga (HVPS ON)	HVPS OFF	HVPS ON
3	HV_DIS_1 (central CEMs HV OFF)	Central CEMs ON	Central CEMs OFF
4	HV_DIS_2 (periphery CEMs HV OFF)	Periphery CEMs ON	Periphery CEMs OFF
5 – 7	Not used	-	-
8	HEATER ON	OFF	ON
9 -13	Not used	-	-
14	HEATER_LOOP_ON	CLOSED LOOP	OPEN LOOP
15 - 23	Not used	-	-

Table 17

The ***getMainControlRegister*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x2800 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the current control register value.

9.2.4. Heater Controller Loop commands

The ***setRegulationCoeff*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2802 16 bits		size=0x0003 16 bits		<pidCoef> 12 bits		12 bits

Argument = <pidCoef> as follows :

Bit	Description
0-11	Kp term
12-23	Ki term

Table 18

The ***getRegulationCoeff*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x2802 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the current Pk and Pi values.

The ***setOpenLoopDuty*** command has the format as follows :

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x5000 16 bits		Size=0x0003 16 bits		<openLoopDuty> 24 bits		

Argument = <openLoopDuty> as follows:

Bit	Description
0-11	Duty cycle from 0x000 to 0xFFE
12	HEATER_OFF (0 is ON, 1 is OFF)

Table 19

The ***setRegulationSetPoint*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2803 16 bits		Size=0x0006 16 bits		<heaterSetup> 24 bits		

Argument = <heaterSetup> as follows

Bit	Description
0-11	Target themperature
12-18	Heater thermometer select, may be 0x05 or 0x06
19-23	Not used

Table 20

The ***getRegulationSetPoint*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x2803 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the current target temperature and used thermometer

9.2.5. *hkRequest* command

The ***hkRequest*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2804 16 bits		size=0x0003 16 bits		0x000001 24 bits		

PAS returns an ACK and then the HK packet.

9.2.6. setPeakTrackingMask commands

The ***setPeakTrackingMask*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x2805 16 bits		size=0x0003 16 bits		mask 24 bits		

Argument = <cemMask> as follows :

Bit	Description
0-10	Channels masks, for the “Count maximum position calculator”. ‘1’ inhibit the corresponding channel, ‘0’ allows that channel
11-23	Not used

Table 21

The ***getPeakTrackingMask*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x2805 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the current mask used.

9.2.7. SetHV_CEM commands

HV_CEM register content must be updated by a pair of commands (enable/go) with the same argument value. HV_CEM may be updated only when Sequencer is running

The ***enableSetHV_CEM*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x00 8 bits	address=0x1800 16 bits		size=0x0003 16 bits		0x000 12 bits	value 12 bits	

Argument = <cemHV> 12 bits value

The ***goSetHV_CEM*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xC0 8 bits	address=0x1800 16 bits		size=0x0003 16 bits		0x000 12 bits	value 12 bits	

The same arguments

9.2.8. SetMAIN_HV commands

MAIN_HV register must be updated by a pair of commands (enable/go) with the same argument value. The MAIN_HV may be updated only when Sequencer is running

The ***enableSetMAIN_HV*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x00 8 bits	address=0x1801 16 bits		size=0x0003 16 bits		0x000 12 bits	value 12 bits	

Argument = <mainHV> 12 bits value

The ***goSetMAIN_HV*** command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xC0 8 bits	address=0x1801 16 bits		size=0x0003 16 bits		0x000 12 bits	value 12 bits	

The same arguments

9.2.1. *clearSharedMemory* command

The command ***clearSharedMemory*** has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	...	...	...	...	Byte 387	Byte 388
ID=0x80 8 bits	address=0x0000 16 bits		Size=0x0180 16 bits		0 128*24 bits								

9.2.1. Sequencer upload and check commands

uploadSequencer command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	...	...	...	...	...	...
ID 8 bits	address 0x800		size 0x1800		Value 0			... Value X ... N x 24 bits			Value N-1		

Argument = <sequerncerFile>
Upload address is 0x800, size is always 0x1800. If the <sequerncerFile> length is less, then DPU shall pad to zero the unused bytes.

getSequencer command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x0800 16 bits		size=0x0003 16 bits		dumpSize=0x001800 24 bits		

Sequencer code shall be received back for debugging purposes. Sequencer has to be disactivated.

9.2.2. ***startSequencer*** command

The command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x0400 16 bits		Size=0x0003 16 bits		0x000001 24 bits		

9.3. List of Configuration Parameters related to the Sequencer Configuration

PAS-ICD-093-001:

DPU shall provide possibility to update the Configuration Parameters shown in Table 23 inside Procedures (Macros) and/or by dedicated telecommands. These parameters are arguments of Sequencer relates native PAS commands.

The following table defines a specific common field in Configuration Parameters. This is NOT a parameter that has to be updated by telecommand, but just a field description and the useful constants that shall be assigned to this field:

Field/constant Name	Value	Description, Comments
<confCEM>	<confCEM_C>	The variable containing CEMs configuration. It corresponds to CHANNELTRON_CONF (addres 0x00) and PKTR_CHANNELTRON_CONF (address 0x10) in Table 15. This variable is not used directly, but is a part of <staticWindow> or <dynamicWindow>. The DPU shall use constants <confCEM_C> and <confCEM_F> to fill <confCEM>. 0x00 – All CEMS are ON 0x01 – Only Periphery CEMs are ON 0x02 – Only Central CEMs are ON

		0x03 – All CEMs are OFF
<confCEM_C>	0x02	Constant to fill the <confCEM> word in <staticWindow> or <dynamicWindow>. Only Central CEMs are ON
<confCEM_F>	0x00	Constant to fill the <confCEM> word in <staticWindow> or <dynamicWindow>. All CEMs are ON

Configuration Parameters Table

Parameter Name	Default Value	Description, Comments
<staticWindow>	{<confCEM_F>,4,92, 0, 9, 1,1}	Variable to save the argument of the command ***loadStaticTable***. Contains 7x24b words. It correspond to “Static scheme parameters table” in Table 23. Can be modified by telecommand. {<confCEM>,Se,Ne,Sel,Nel, K, N} (low word first)
<dynamicWindow>	{<confCEM_F>, 4, 92, 0, 9, 48, 3, 4, 19, 15}	Variable to save the argument of the command ***loadDynamicWindowTable***. Contains 7x24b words. It correspond to “Dynamic Window scheme parameters table” in Table 23. Can be modified by telecommand. {<confCEM>,SeF3D, NeF3D, SelF3D, NelF3D,Ne,Nel,K,N,L}
<testGenerator>	{0, 40, 5, 5, 20}	Variable to save the argument of the command ***loadPulsesTable.*** It corresponds to “Pulses Injection” in Table 23. Can be modified by telecommand. See important requirement in 9.5.15 {enbl, ie, iel, icem, cnt} { 0/1, energy 0 – 95, elevation 0 – 8, CEM 0 – 10, count 0 – 30}
<enginSampling>	{0xE9EB0A, 0x6D703E, 0x0FBD74, 0x0FBD74, 0x1F6F0F, 0x00001E} #{1280V, 0.4275, 0.9838, 0.9838, 1.965, 30s}	Variable to save the argument of the command ***loadEngineeringTable***. It correspond to “Pulses Injection” in Table 23. Can be modified by telecommand. See important requirement in 9.5.12 { Anl_Max, K, Anl_TD_Ration, Anl_BD_Ratio, Anl_TC_Ratio, HT_Step_Dur}
<confStepping>	TBD	The array 60 x 24b words. The variable to save the argument of the command ***loadConfigurationTable*** Can be modified by telecommand. See important requirement in 9.5.11

Table 22

9.4. PAS FPGA shared memory address mapping

The following table describes the memory mapping of each Sequencer constant described in **PAS-R-057-14** (RD-1). Somme fields correspond to DPU configuration parameters, see Table 22.

Address		Parameter	Comment
Static scheme Parameters table	0x00	CHANNELTRON_CONF = <confCEM>	Channeltrons to use: 0 = all, 1 =lateral, 2=central, 3=none
	0x01	FIRST_ENERGY = Se	Start energy (0..95)
	0x02	ENERGY_NUMBER = Ne	Energy nb (1..96)
	0x03	FIRST_ELEVATION = Sel	Start elevation (0..8)
	0x04	ELEVATION_NUMBER = Nel	Elevation nb(1..9)
	0x05	K_VALUE = K	nb samplings per second
	0x06	N_VALUE = N	nb seconds
Not used	0x07 to 0x0f		
Dynamic window scheme parameters table	0x10	PKTR_CHANNELTRON_CONF = <confCEM>	Channeltrons to use: 0 = all, 1 =lateral, 2=central, 3=none
	0x11	PKTR_PRESCAN_FIRST_ENERGY = 0	Full3D Start energy = 0
	0x12	PKTR_PRESCAN_ENERGY_NUMBER = 96	Full3D Energy number = 96
	0x13	PKTR_PRESCAN_FIRST_ELEVATION = 0	Full3D Start elevation = 0
	0x14	PKTR_PRESCAN_ELEVATION_NUMBER = 9	Full 3D Elevation = 9
	0x15	PKTR_ENERGY_WINDOW_SIZE = Ne	Energy window size (1..96)
	0x16	PKTR_ELEVATION_WINDOW_SIZE = Nel	Elevation window size (1, 3, 5, 7, 9)
	0x17	PKTR_K_VALUE = K	nb samplings per second
	0x18	PKTR_N_VALUE = N	nb seconds
	0x19	PKTR_L_VALUE = L	Number of repetitions of sequence
Engineering parameters table	0x1a	ENG_HT_ANL_MAX	tension de depart = (Uanl max /coeff anl) * 2^7
	0x1b	ENG_K	K = exp(log(Umax/ Umin)/3) => K * 2^24
	0x1c	ENG_ANL_TD_RATIO	ANL_TD_RATIO = (Utdef max / coeff tdef) / (Uanl max / coeff anl) * 2^20
	0x1d	ENG_ANL_BD_RATIO	ANL_BD_RATIO = (Ubdef max / coeff bdef) / (Uanl max / coeff anl) * 2^20
	0x1e	ENG_ANL_TC_RATIO	ANL_TC_RATIO = (Ucap max / coeff cap) / (Uanl max / coeff anl) * 2^20
	0x1f	ENG_HT_STEP_DURATION	N secondes (multiple de 5s) DEFAUT = 30s
Pulses Injection table	0x20	INJECT_PULSES_ENABLED	Enable pulse injection: 0xF1A6E => On, else=Off
	0x21	INJECT_PULSES_BIN = ie,iel coded : Bits 23..11: spare Bits 10..4: Ie (energy) Bits 3..0: Iel (elevat)	Coded Energy and elevation bin to inject pulses

	0x22	INJECT_PULSES_CHANNELTRON_NUM = icem	Channeltron number to inject pulses
	0x23	INJECT_PULSES_COUNT = cnt	Pulses nb (MAX 30)
Stepping configuration table	0x24 to 0x5F	60 words of configuration table. The internal layout TBD	
Locked memory by FPGA	0x60	SEQ_ID	
	0x61	HDR_PKT_MODE1	
	0x62	HDR_PKT_MODE2	
	0x63		
	0x64	SCI_PKT_DATA_FIELD_SIZE	
	0x65	SCI_PKT_ENERGY_BIN	
	0x66	PKTR_RESULT	
Not used	0x67 To 0x7a		
Mailboxes DPU<->Seq	0x7b	MAILBOX_SEQ_DEBUG	**seq->DPU** : debug informations
	0x7c	MAILBOX_VERSION	**seq->DPU** : sequencer version
	0x7d	MAILBOX_SEQ_ERROR	**seq->DPU**: error code : 0 => no error 1 => ERR_BAD_ENERGY_PARAMETERS 2 => ERR_BAD_ELEVATION_PARAMETERS 3 => ERR_BAD_PKTR_WINDOW_SIZE 4 => ERR_BAD_K_PARAMETER
	0x7e	MAILBOX_FROM_DPU	**DPU-> seq** : mailbox commands from DPU 0 => DO_NOTHING (IDLE mode) 1 =>DO_STATIC_SAMPLING 2 =>DO_DYNAMIC_WINDOW_SAMPLING 3 => DO_ENGINEERING_SAMPLING 0xFF =>ABORT_SEQUENCE
	0x7f	MAILBOX_SEQ_STATE	**seq->DPU** : Sequencer state 0 => DO_NOTHING (Idle) 1 =>DO_STATIC_SAMPLING 2 => DO_DYNAMIC_WINDOW_SAMPLING 3 => DO_ENGINEERING_WINDOW

Table 23

9.5. PAS Sequencer commands

Sequencer Command is a command that DPU sends to PAS to update Sequencer Configuration.

9.5.1. Commands list

PAS-ICD-095-001:
DPU shall provide Sequencer Commands as shown in Table 24.

Command	Argument	Description	Condition	Return
getSequencerVersion		Get the version of the running sequencer	Sequencer must be running	Data requested
getSequencerState		Get the state of the sequencer	Sequencer must be running	Data requested
getSequencerDebugValue		Used for sequencer debug	Sequencer must be running	Data requested
getSequencerError		Get the current Error encountered by the sequencer	Sequencer must be running	Data requested
abortExecution		If the sequencer is in IDLE mode, the command has no effect; else the sequencer finishes properly the current execution at the next ½s tick and come back to IDLE mode.	Sequencer must be running	ACK/NACK
execEngineeringSampling ***execStaticSampling*** ***execDynamicWindowSampling***		Set a flag to indicate to the sequencer to enter to sampling mode and execute a specific scheme. If the sequence is in IDLE mode, the sampling mode is entered at the next ½s tick; else the new scheme will start when the current scheme execution will be	Sequencer must be running	ACK/NACK

		completely finished. To come back to Idle mode, the ***abortExecution*** or ***gotoIdleMode*** command must be used.		
gotoIdleMode		If the sequencer is in IDLE mode, the command has no effect; else the sequencer finishes the current scheme execution and come back to IDLE mode.	Sequencer must be running	ACK/NACK
loadConfigurationTable	<confStepping>	Loads Sequencer configuration (HV stepping constants)	Sequences shall be in IDLE mode	ACK/NACK
loadEngineeringTable	<enginSampling>	Loads parameters table used to control engineering scheme	See 10.1.2	ACK/NACK
loadStaticTable	<staticWindow>	Loads parameters table used to control Static scheme	See 10.1.2	ACK/NACK
loadDynamicWindowTable	<dynamicWindow>	Loads parameters table used to control Dynamic Window scheme	See 10.1.2	ACK/NACK
loadPulsesTable	<testGenerator>	Loads parameters table to control the test pulses generation for test putposes	See 10.1.2	ACK/NACK
shareMemoryDump	-	Dump the entire contents of the shared memory	Sequencer at any state	Data requested

Table 24

9.5.2. ***getSequencerVersion*** command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x007C 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the sequencer version:

Bit	Version description
0-3	Subversion number

Bit	Version description
4-7	Version number
8-23	Sequencer type : 0x1 => test sequencer (used to debug PAS board) 0x2=> science sequencer

9.5.3. *getSequencerState* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x007F 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the sequencer state:

Bit	State description
0-23	State : 0 => is in IDLE mode 1 => is executing Static scheme in sampling mode 2 => is executing Dynamic Window scheme in sampling mode 3 => is executing Engineering scheme in sampling mode

9.5.4. *getSequencerDebugValue* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x007B 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the sequencer state:

Bit	Debug information description
0-23	Used to debug the sequencer, TBD

9.5.5. *getSequencerError* command

The command has the following format:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x007D 16 bits		Size=0x0003 16 bits		dumpSize=0x000003 24 bits		

PAS returns the error encountered:

Bit	Error description

Bit	Error description
0-23	Error code: 0 => no errors 1 => bad energy parameters 2 => bad elevation parameters 3 => bad peaktracking window size 4 => bad K parameter

9.5.6. *abortExecution* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x007E 16 bits		Size=0x0003 16 bits		0x0000FF 24 bits		

9.5.7. *execEngineeringSampling* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6		Byte 7
ID=0x80 8 bits	address=0x007E 16 bits		size=0x0003 16 bits		0x00 8 bits	info 4 bits	0x003 12 bits	

DPU shall provide the SCIENTIFIC MODE index in bits 12…15 ("info"). PAS FPGA copies this index to the PAS Scientific Header. See Table 3 and comments below the table for details.

See 10.1.1, 10.1.2 and PAS-ICD-101-001 for command usage.

9.5.8. *execStaticSampling* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6		Byte 7
ID=0x80 8 bits	address=0x007E 16 bits		size=0x0003 16 bits		0x00 8 bits	info 4 bits	0x001 12 bits	

DPU shall provide the SCIENTIFIC MODE index in bits 12…15 ("info"). PAS FPGA copies this index to the PAS Scientific Header. See Table 3 and comments below the table for details.

See 10.1.1, 10.1.2, and PAS-ICD-101-001 for command usage.

9.5.9. *execDynamicWindowSampling* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6		Byte 7
ID=0x80 8 bits	address=0x007E 16 bits		size=0x0003 16 bits		0x00 8 bits	info 4 bits	0x002 12 bits	

DPU shall provide the SCIENTIFIC MODE index in bits 12…15 ("info"). PAS FPGA copies this index to the PAS Scientific Header. See Table 3 and comments below the table for details.

See 10.1.1, 10.1.2, and PAS-ICD-101-001 for command usage.

9.5.10. *gotoIdleMode* command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0x80 8 bits	address=0x007E 16 bits		Size=0x0003 16 bits		0x000000 24 bits		

9.5.11. *loadConfigurationTable* command

The command uploads the Stepping High Voltage configuration table to memory. The size of the table is 60x24bits of size

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	...	...	...	Byte 62	Byte 63	Byte 64
ID=0x80 8 bits	address=0x0024 16 bits		Size=0x00B4 16 bits		configurationTable 60*24 bits								

Argument = <confStepping>, see Table 22

IMPORTANT: This command can be executed in the FPGA IDLE state ONLY. To do this DPU shall :

1. Perform ***gotoIdleMode*** command
2. Periodically perform ***getSequencerState*** command until the response is 0
3. Perform ***loadConfigurationTable*** <confStepping> command

9.5.12. l*oadEngineeringTable* command

The command uploads engineering table to memory.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	...	...	...	Byte 20	Byte 21	Byte 22
ID=0x80 8 bits	address=0x001a 16 bits		Size=0x00012 16 bits		engineeringtable 6*24 bits								

Argument = <enginSampling>, see Table 22. See PAS-ICD-051-00, 10.1.2 and PAS-ICD-101-001 for command usage.

9.5.13. *loadStaticTable* command

The command uploads static scheme parameters table to memory.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13
ID=0x80 8 bits	address=0x0000 16 bits		Size=0x00015 16 bits		channeltronCconf 24 bits			startEnergy 24 bits			nbEnergies 24 bits		

Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25
startElevation 24 bits			nbElevations 24 bits			K 24 bits			N 24 bits		

Arguments = <staticWindow>, see table 22. See PAS-ICD-051-001, 10.1.2, and PAS-ICD-101-001 for command usage.

9.5.14. *loadDynamicWindowTable* command

The command uploads dynamic scheme parameters table to memory.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte 16
ID=0x80 8 bits	address=0x0010 16 bits		Size=0x0001E 16 bits		channeltronConf 24 bits			startEnergy 24 bits			nbEnergies 24 bits			startElevation 24 bits		

Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31	Byte 32	Byte 33	Byte 34
nbElevations 24 bits			energyWindowSize 24 bits			elevationWindowSize 24 bits			K 24 bits			N 24 bits			L 24 bits		

Arguments = <dynamicWindow>, see table 22. See PAS-ICD-051-001, 10.1.2, and PAS-ICD-101-001 for command usage.

9.5.15. *loadPulsesTable* command

The command uploads pulses definition to memory.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte 16
ID=0x80 8 bits	address=0x0020 16 bits		size=0x000C 16 bits		onOFF 24 bits			bin 24 bits			channeltronNum 24 bits			pulseCount 24 bits		

Argument = <testGenerator>, see Table 22. Coding of the bytes 8..10 see in Table 24 (field INJECT_PULSES_BIN). See PAS-ICD-051-001 for command usage.

9.5.16. *shareMemoryDump* command

The command get back the full shared memory contents.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID=0xA0 8 bits	address=0x0000 16 bits		size=0x0003 16 bits		dumpSize=0x000180 24 bits		

The Sequencer can be in any state

9.6. Time control

The operation of the PAS time stamping system is almost identical to the one used at spacecraft level, as defined in the EID-A.

The FPGA's time control system performs the following functions:

1. Provides centralised, internal synchronisation and timing
2. Provides time information for the science and HK packets time stamping

The reason this is required is that the DPU receives science and HK data packets from the instrument and stores them in a FIFO; the delay in the FIFO is indeterminate so the DPU is unable to time stamp received data. Therefore EAS and PAS must be able to time stamp their own packets.

Resolution:

The time is transmitted as the SET (CUC) as defined in the EID-A. The local clock is 19.98848MHz, this gives the finest resolution that can be obtained in the CUC format used by the spacecraft, so all 48 bits are valid.

Bits 47 – 40	Bits 39 – 32	Bits 31 – 24	Bits 23 – 16	Bits 15 – 8	Bits 7 – 0
VVVVVVVV	VVVVVVVV	VVVVVVVV	VVVVVVVV	VVVVVVVV	VVVVVVVV

'V' = valid

The time controller is mapped as 2 locations, which are accessed as memory access type commands.

The locations are:

Time coarse bits (24 MSBs): 0x3000, write and read, this location must be written and read first (as it will be if the time is uploaded at a single six byte data write/read packet)

Time fine bits: 0x3001, write and read.

Time control register: 0x3002. Currently not implemented.

A time update command is sent to PAS which identifies the time as it will be when the next SpW time code is sent. I.e. the time controller saves the new value in a 48 bit register until the next time code arrives. A missing time code just allows the time counter to continue. The value in the SpW time code is unused.

9.6.1. Time management

The PAS time management system provides the time stamping for science header and HK packets. This system runs indipendantly from the sequencer. The following write packet should be sent every 1s, 55napsho. 300ms before the 1s boundary, followed by a SpW time code on the 1s boundary

9.6.2. Uploading/downloading System Time

The time (at the next SpW time code receipt) shall be uploaded as a single write type command packet, carrying two 24 bit values. It could be read back in the same fashion, using a memory read type command, with a data length field of 6 (two 24 bit words). It is unlikely that reading back the value by the DPU will be used except in ground testing, as it reads back the current system time, not the new uploaded time.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
ID=0x80 8 bits	address=0x3000 16 bits		Size=0x0006 16 bits		time 48 bits					

Issue SpW time code at the next 1s boundary

10. PAS – DPU synchronization requirements

10.1. DPU commanding synchronization

10.1.1. Sequencing of sequences

The DPU commands the scientific operation of PAS by sending parameter tables and updating the DPU MAILBOX content. The procedure is described in the next subsections. We call "**the sequence**" a specified Sequencer scheme executing with a specified parameter table. Note that the sequence duration is defined in the parameter table. The minimal duration is one sec. When PAS sequencer has finished the current sequence (see Figure 6) it waits 1s before starting the next sequence. The DPU can send new parameters tables to PAS during the current sequence execution and before the end of the 1s wait. Then PAS will operate accordingly, with a new scheme. If no table has been sent, and DPU MAILBOX has not been changed, PAS repeats the current sequence.

Note that the new parameter table can be sent anytime, but PAS will ignore it when measuring.

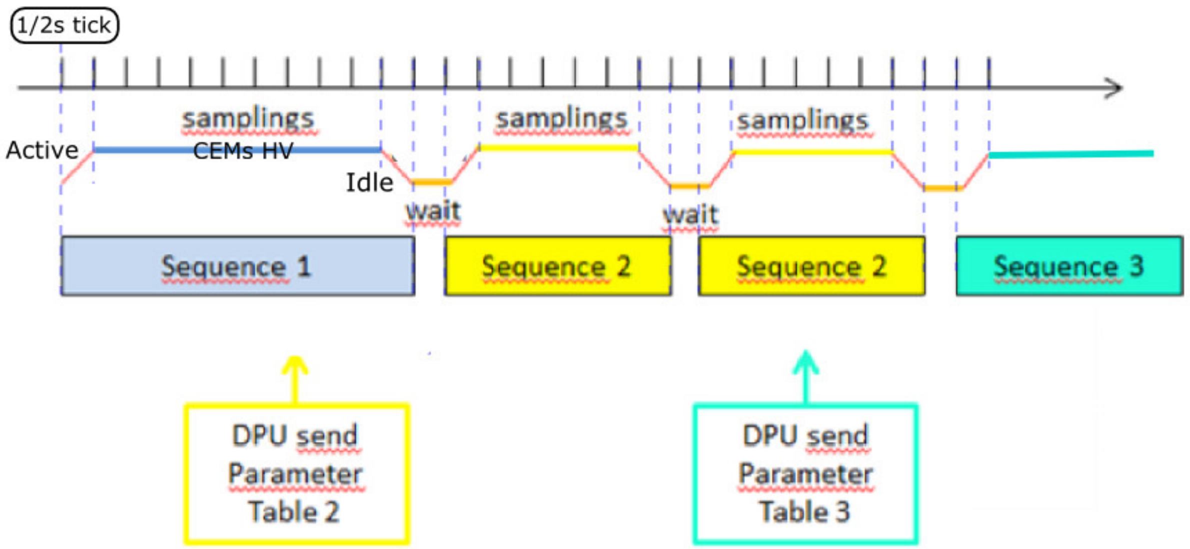


Figure 5 DPU commands processing diagram

10.1.2. Sequences tables configuration

The access to sequences tables is secured by a mutual exclusion mechanism.

When the sequencer is ready to start a new sequence (in the idle mode or during the 1s wait between two sequences), it preforms as follows:

1. reads the **MAILBOX_FROM_DPU**, if a value indicates to enter in sampling (static, dynamic window or engineering scheme)
2. loads the corresponding parameters table,
3. reads again the **MAILBOX_FROM_DPU**

If the **MAILBOX_FROM_DPU** contents has not been changed, Sequencer enters the sampling mode and starts the sequence execution, if the value has been changed, it continues in idle mode.

The mutual exclusion process (1)(2)(3) duration is less than 1 ms (around 200 µs in the rality).

PAS-ICD-101-001: PAS Sequencer parameter table updating and start a sequence

DPU shall use the protocol described in PAS-ICD-051-001 to update the Sequencer parameter tables such as «Engineering Parameters Table», «Static Scheme Parameters table», and «Dynamic Scheme Parameters Table» (see Table 24). When DPU is going to update such a table, it shall firstly send ***gotoIdleMode*** command, then update the table (***loadEngineeringTable***, ***loadStaticTable***, and ***loadDynamicWindowTable*** commands), and then **wait 1ms minimum** before sending a ***execXXXSampling*** command. The 1ms delay ensures the mutual exclusion. (DPU shall do not execute this commands sequence during the first 1 ms time interval of each second (considering that the second starts with **½s** FPGA tick).

Note that the for Configuration tables ("Pulse Injection Table" and "Stepping Table") upload DPU shall use requirements PAS-ICD-051-001 and PAS-ICD-095-001.

10.1.3. Normal Mode Synchronisation

PAS-ICD-101-002: PAS Normal mode commanding

To synchronize PAS schemes performance in the NORMAL mode with 4sec boundary, the commands to start execution of corresponding scheme (***execXXXSampling)*** shall be sent as shown in Figure 6.

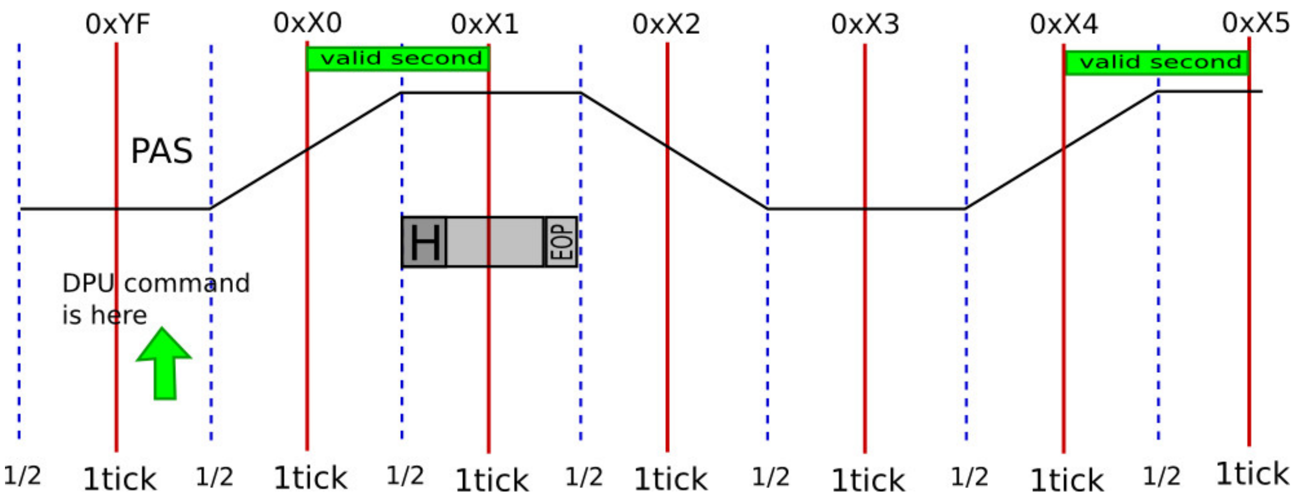


Figure 6 PAS synchronization in NORMAL mode

11. How DPU control Modes and setup PAS

From DPU data products point of view there are tree modes of PAS:

NORMAL mode where DPU receive one Sampling from PAS per 4 s. And every 300 s the short fast Dynamic Scheme interval (SnapShot) is activated. See section 13.3.1 for details.

BURST mode is a maximal 300 sec fast data production time interval when PAS in working in the Dynamic Window Scheme. The PAS data are storied in a ring buffer. See section 13.3.2 for details.

ENGINEERING mode where PAS is working in any Scientific Scheme or Engineering scheme with no data compression for maximal 300 sec. Data are storied in the ring buffer as well us BURST mode data.

Also there is a special PAS activity called "**CALIBRATION**". DPU shall perform the dedicated procedure (see section 13.3), put the data into ring buffer (as well as for the BURST mode), process the data and transfer the data as low latency data (see section 7.4).

From the command point of view any mode is the performance a sequence of sequences as described in 10.1.1 and Figure 5. The cyclograms (see Section 13) controls the Modes execution. Before modes execution PAS has to be set a proper configuration. In addition, if an error is found, some emergency macros have to be performed. The full diagram of PAS functional cycle is shown in Figure 7.

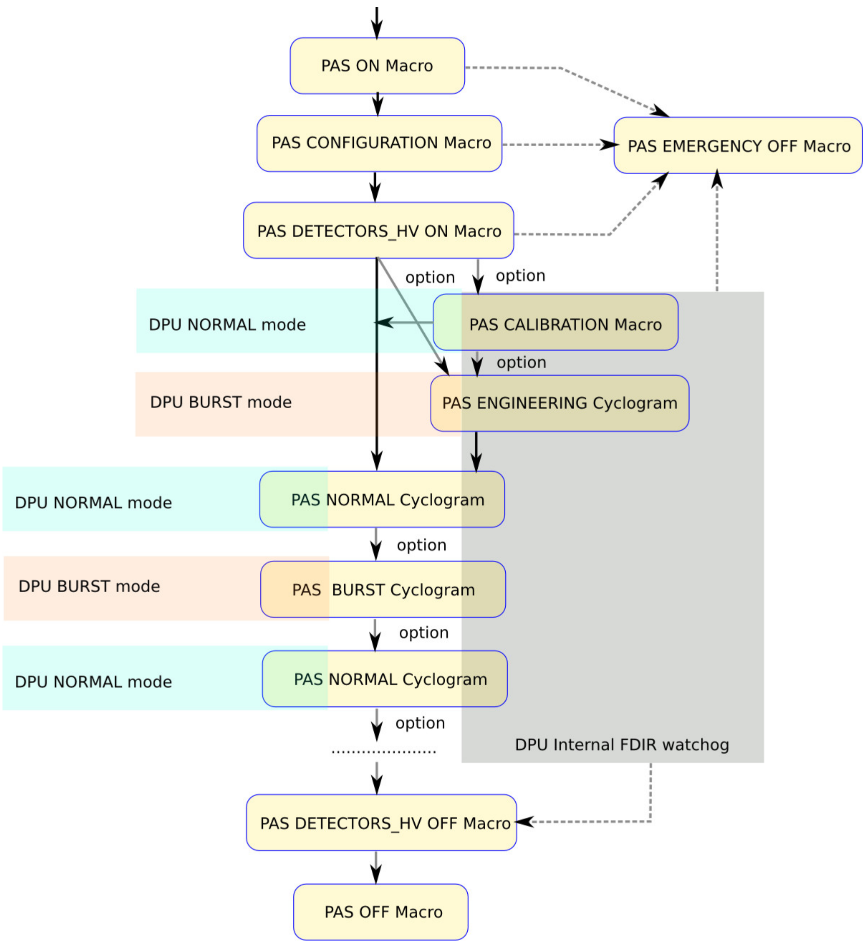


Figure 7 Full diagram of PAS activity cycle

12. Configuration procedures

PAS procedures are the command macros devoted to set PAS to some running state

12.1. Procedures Related Configuration Parameters

PAS-ICD-111-001:

DPU shall keep the configuration parameters as shown in Tables 14, 22 and Table 25. Each Configuration Parameter shall be modifiable by dedicated telecommand.

Parameter	Default value	Description or reference
<cemHVnominal>	0x5C4	The nominal HV = 1800V. Can be changed by telecommand
<stepCEM>	0xA4	Step of CEM HV ramping = 200V Can be changed by telecommand
<minCalibrCEM>	0x3D8	Start CEM HV for calibration = 1200V
<maxCalibrCEM>	0x6BA	Stop CEM HV for calibration = 2100V
<stepCalibrCEM>	0x029	Step of CEM HV for calibration = 50V Can be changed by telecommand
<I_28V_Init_MIN>, <I_28V_Init_MAX>	0x02bc, 0x0349	PAS consumption limits in "standby" state.
<I_p12V_Init_MIN>, <I_p12V_Init_MAX>	0x038b, 0x049b	PAS +12 V current limits in in "standby" state.
<I_m12V_Init_MIN>, <I_m12V_Init_MAX>	0x019b, 0x0135	PAS -12 V current limits in in "standby" state
<I_3V3_Init_MIN>, <I_3V3_Init_MAX>	0x05ca, 0x0703	PAS +3.3 V current limits in in "standby" state
<I_1V5_Init_MIN>, <I_1V5_Init_MAX>	0x05e6, 0x0715	PAS +1.5 V current limits in in "standby" state

Table 25 Additional Configuration Parameter List (see Tables 11 and 22 also)

12.2. DPU commands/telecommands related to the Procedures

PAS-ICD-111-002:

To support the PAS procedures DPU shall support at least commands/ telecommands shown in Table 26. This list does not include any assignment operation.

Command Name	Command Description
calculatePeakPosition	Perform calculations explained in Section 6.5 Set Se and Sel into <staticWindow> Configuration Parameter
transmitPeakPosition	Generate data product PAS_CALIBR (Section 7.1)

Table 26

12.3. Configuration procedures

PAS-ICD-111-003:

DPU shall implement at least Cofiguration Procedures indicated in Table 27 and described in subsection

Procedure name	Procedure description
PAS_ON	General operations before PAS configuration
PAS_CONFIGURATION	General PAS setup
PAS_OFF	Long and safe PAS OFF
PAS_DETECTORS_HV_ON	Ramp up the CEMs
PAS_DETECTORS_HV_OFF	Ramp down CEMs
PAS_EMERGENCY_OFF	Fast OFF

Table 27

12.3.1. Implementation of ***PAS_ON***

See the enclosed file “PAS_Macro_ON_25_Jul_2017.xlsx”

12.3.2. Implementation of ***PAS_CONFIGURATION***

See the enclosed file “PAS_Macro_CONFIGURATION_21_Jun_2017.xlsx”

12.3.3. Implementation of ***PAS_OFF***

See the enclosed file “PAS_Macro_OFF_21_Jun_2016.xlsx”

12.3.4. Implementation ***PAS_DETECORS_HV_ON***

See the enclosed file “PAS_Macro_DETECTORS_ON_25_Jul_2017.xlsx”

12.3.5. Implementation ***PAS_DETECTORS_HV_OFF***

See the enclosed file “PAS_Macro_DETECTORS_OFF_21_Jun_2017.xlsx”

12.3.6. *Implementation PAS_EMERGENCY_OFF*

See the enclosed file “PAS_Macro_Emergency_OFF_21_Jun_2017.xlsx”

13. PAS cyclograms

PAS snapshots is possibly looping procedure that perform PAS mode execution.

13.1. Cyclogram dedicated Configuration Parameters

PAS-ICD-131-001:

To support the PAS cyclograms DPU shall keep at least Configuration Parameters identified in Table 28. Each Configuration Parameter shall be modifiable by dedicated telecommand. These parameters are used to replace the schemes configuration arguments in corresponding commands (like ***loadDynamicWindowScheme***, etc)

Parameter Name	Default value	Description
<dynamicSnapshot>	{<confCEM_F>,4, 92, 0, 9, 48, 3, 4, 7, 1}	{CEM_conf, SeF3D, NeF3D, SelFeD, Ne, Nel, K, N, L} Dynamic Window scheme parameters to load for the snapshot (see also table 22)
<normalSampling>	{<confCEM_F>,24, 48, 1, 5, 1}	{ CEM_conf, Se, Ne, Sel, Nel, K,N} Static Scheme parameters for the NORMAL mode (see also table 22)
<normalFull3D>	Default <NormalWindow> = {<confCEM_F>,4,92,0,9,1,1}	{ CEM_conf, Se, Ne, Sel, Nel, K,N} Static Scheme Full3D for the Normal mode (see also table 22)
<dynamicBurst>	{<confCEM_F>,4, 92, 0, 9, 48, 3, 4, 19, 15}	{CEM_conf, SeF3D, NeF3D, SelFeD, Ne, Nel, K, N, L} Dynamic Window for the BURST mode (see also table 22)
<staticWindowCalibration>	{<confCEM_F>,4,92, 0, 9, 1,1}	Static window for calibration purposes. {<confCEM>,Se,Ne,Sel,Nel,K,N} (low word first) (see also table 22)
<CEMcalibrHVStart>	0	HV (digital reference) CEM HV value to start calibration
<CEMcalibrHVStep>	0	HV (digital reference) CEM HV step value for calibration
<CEMcalibrHVStop>	0	HV (digital reference) CEM HV value to stop calibration
<CEMcalibrHVnom>	0	HV (digital reference) CEM HV value to set when calibration is completed

<CEMcalibr_mask_0>	0x02	CEM #1
<CEMcalibr_mask_1>	0x04	CEM #2
<CEMcalibr_mask_2>	0x08	CEM #3
<CEMcalibr_mask_3>	0x10	CEM #4
<CEMcalibr_mask_4>	0x20	CEM #5

Table 28

Also following variables are used in the cyclograms:

Variable	Init value	Description
<Sample_Number>	0	CEM calibration index corresponding to the step number of the CEM_HV
<maxCountPosition>		See p.33

Table 29

Cyclogram dedicated Commands

PAS-ICD-132-001:

To support the PAS cyclograms DPU shall support at least commands/ telecommands shown in Table 30.

Command Name	Command arguments	Command Description
startNORMAL_mode	-	Start the Cyclogram
startBURST_mode	-	
startCALIBRARION		CALIBRATION
startENGENEERING		ENGINEERING Mode
stopCyclogram	-	Stop any cyclogram
defineSnapshot	CEM_config, Se, Ne, Sel, Nel for initial Full3D Ne,Se,K,N,L for Dynamic Scheme	Set values to <dynamicSnapshot>
defineBurst	CEM_config, Se, Ne, Sel, Nel for initial Full3D Ne,Se,K,N,L for Dynamic Scheme	Set values to <dynamicBurst>
defineNormal	CEM_config, Se, Ne, Sel, Nel, K for static scheme	Set values to <normalSampling>
defineFS	CEM_config, Se, Ne, Sel, Nel	Set values to <normalFull3D>
startCEMcalibr	CEMcalibrHVStart,	Define all necessary

	CEMcalibrHVStep, CEMcalibrHVStop, CEMcalibrHVnom, CEMcalibr_mask_0, CEMcalibr_mask_1, CEMcalibr_mask_2, CEMcalibr_mask_3, CEMcalibr_mask_4	parameters and start CEM calibration cyclogram

Table 30

13.2. Cyclograms implementation

PAS-ICD-123-001:

DPU shall provide cyclograms as follows:

- **NORMAL_mode** – Normal mode with dynamic snapshots. See the corresponsing cyclogram text in subsection 13.1.1. PAS sampling header shall arrives exactly inside a coarse Four sec. Every 100 sec the PAS performs Full Sampling in Static Scheme according to <normalSampling> set, and DPU shall calculate the new Static Window and update the Static Scheme parameters. Every 300 sec DPU shall command exectution of 9 sec of Dynamic Window Scheme as it is defined in <dynamicSnapshot> and, tehn, calculate new static window and restart the Static Scheme according to <normalSampling>. See Figure 9 for detail. **Note:** The Full3D sampling and "Snap Shots" are shifted from the 100s and 300 boundaries due to DPU-PAS communication features.

- **BURST_mode** – Burst mode is the limited (5min) duration Dynamic Window Scheme. See the corresponding cyclogram in subsection 13.3.2. The data of this mode shall be storied in a ring buffer. If DPU see that the PAS data production is not finished, but the time interval specified specified for the bust mode is finished, DPU shall sent the command ***"abortExecution"*** and start to perform the dummy static schema waiting 300s boundary sysnchronization.

- **ENGINEERING_mode** – this mode is dedicated to test any PAS functionality. Thus there is no a special cyclogram for such mode. PAS can execute any scheme with any parameters for a short (5min maximum) duration. The data shall be storied in the ring buffer (as well as the for BURST mode).

- **CALIBRATION** – This mode perform fast measurement varying the CEMs HV. Date shall be storied in the rolling buffer end then processed.

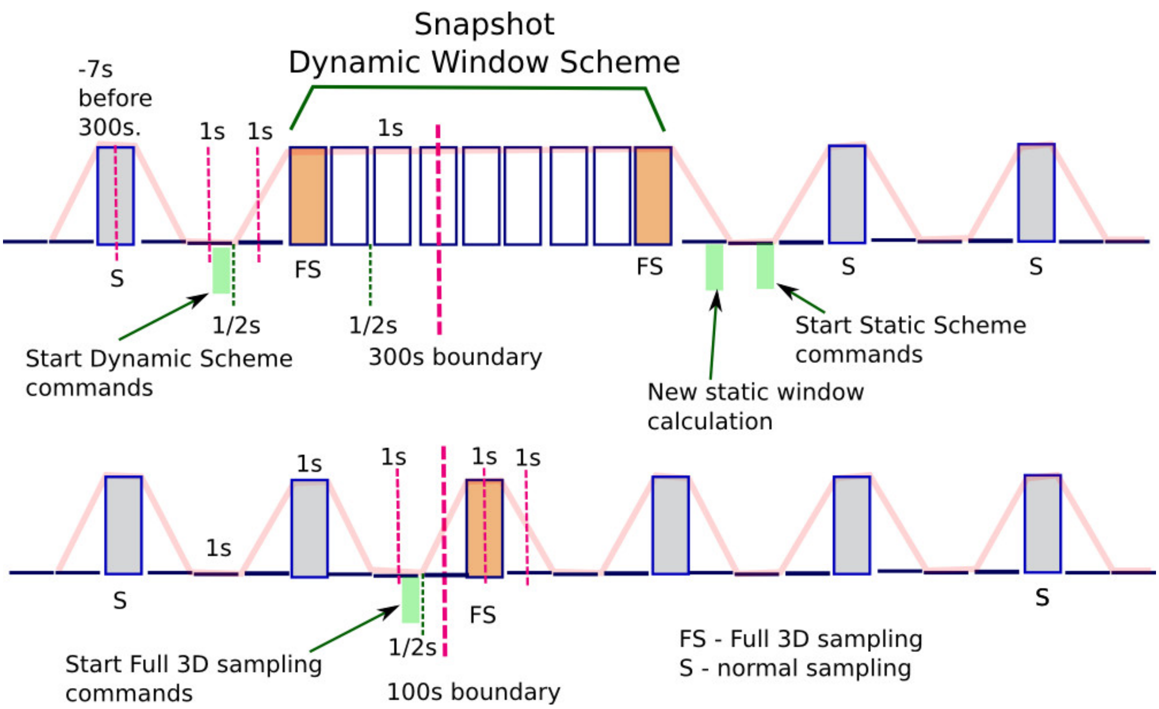


Figure 8 scenario of the Normal Mode cyclogram

13.2.1. <u>Implementation of **NORMAL_mode**</u>

See the enclosed file “PAS_Normal_mode_Cyclogram_V_2_3_29_Aug_2017.xlsx»

13.2.2. <u>Implementation of **BURST_mode**</u>

See the enclosed file “PAS_Burst_mode_Cyclogram_V_2_0_05_Oct_2016.xlsx”

13.2.3. <u>CALIBRATION cyclogram</u>

See the enclosed file “PAS_Calibration_cyclogram_06_Apr_2017.xlsx”

14. ANNEX. The Moment Calculation Constants

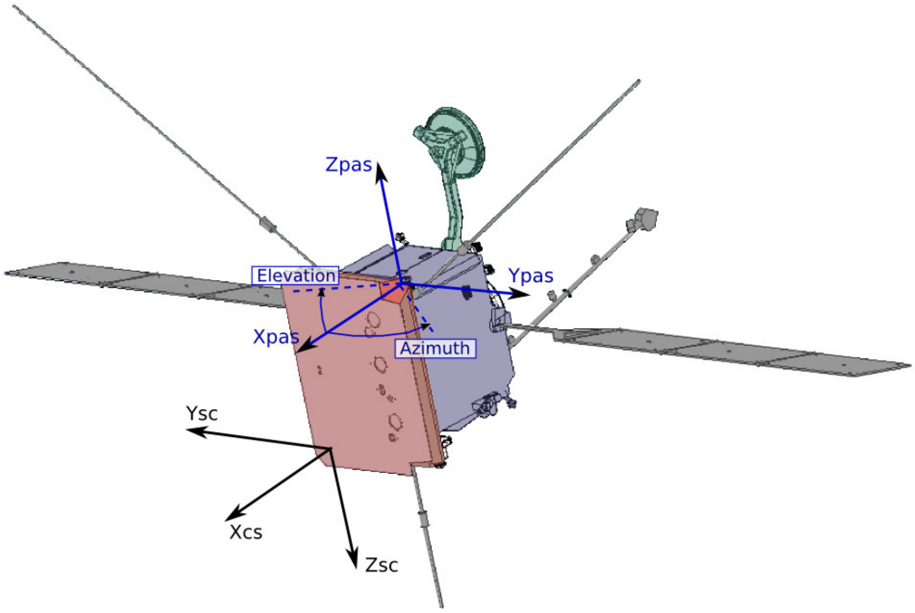
See inserted PDF file below.

PAS approach to the Moment calculation

A. Fedorv, 02/14/2017

This text describes the method how PAS team definess constants needed to ion moment calculations.

1. The PAS frame definition



2. The parameters defined during PAS calibration.

The parameters defined in calibration process are as follows:

Aie, iel, iaz Effective aperture, cm^2, for each energy, $ie = [0...95]$, each elevation, $iel = [0...8]$, and each azimuthal direction, $iaz = [0...10]$. Thus this is an array $96 \times 9 \times 11$. Note that these values already contains cosines since they are the results of the measurement. Also this values contains the detector efficiency as a function of the energy and individual CEM properties. **Important note:** These values are subject of update in flight!

$\Delta E_{ie,iel,iaz}/E$ Energy resolution, unit-less. It is also an array $96 \times 9 \times 11$. These values also can be updated in flight if we are going to modify the stepping voltages in the PAS sequencer.

$\Omega_{ie,iel,iaz}$ Solid angle resolution, sr. It is also an array $96 \times 9 \times 11$. These values also can be updated in flight if we are going to modify the stepping voltages in the PAS sequencer.

$E_{ie,iel,iaz}$ Energy per charge, eV. It is also an array $96 \times 9 \times 11$. These values also can be updated in flight if we are going to modify the stepping voltages in the PAS sequencer.

$Elev_{ie,iel,iaz}$ Elevation angles, rad. It is also an array $96 \times 9 \times 11$. These values also can be updated in flight if we are going to modify the stepping voltages in the PAS sequencer.

Az_{iaz} Azimuth angles, rad. It is an array of 11 values

From the parameters directly obtained during the calibration we calculate (by rather sophisticated procedure) a very important parameter:

$\Delta W_{ie,iel,iaz}$ the integrating velocity volume, $cm/s \cdot sr \cdot (cm/s)^2$, this volume defined by the measurement scheme rather than the calibration.

3. Theoretical part: How to calculate the distribution function and the "partial density" from the count rate

$DF_{ie,iel,iaz}$ ion distribution function, $cm^{-3}(cm/s)^{-3}$ at the points $[V^X, V^Y, V^Z]_{ie,iel,iaz}$

$$Count_{ie,iel,iaz} = DF_{ie,iel,iaz} \cdot V_{ie,iel,iaz} \cdot A_{ie,iel,iaz} \cdot \Omega_{ie,iel,iaz} \cdot V^2_{ie,iel,iaz} \cdot \Delta V_{ie,iel,iaz}/V \cdot V_{ie,iel,iaz} =$$
$$DF_{ie,iel,iaz} \cdot V^4_{ie,iel,iaz} \cdot GV_{ie,iel,iaz}$$

Here:

$GV_{ie,iel,iaz} = A_{ie,iel,iaz} \cdot \Omega_{ie,iel,iaz} \cdot \Delta V_{ie,iel,iaz}/V$ "velocity" geometrical factor, $cm^2 sr$

$\Delta V_{ie,iel,iaz}/V = \Delta E_{ie,iel,iaz}/E/2$ resolution in the velocity space

Thus the distribution function is calculated as:

$$DF_{ie,iel,iaz} = Count_{ie,iel,iaz}/(GV_{ie,iel,iaz} \cdot V^4_{ie,iel,iaz})$$

To calculate moments we have to assume that the $DF_{ie,iel,iaz} = const$ inside the corresponding $\Delta W_{ie,iel,iaz}$ and integrate it. The unity of integration is the partial density $\Delta n_{ie,iel,iaz}$ which is the integral of the distribution function over $\Delta W_{ie,iel,iaz}$:

$$\Delta n_{ie,iel,iaz} = DF_{ie,iel,iaz} \cdot \Delta W_{ie,iel,iaz}, [cm-3]$$

thus

$$\Delta n_{ie,iel,iaz} = Count_{ie,iel,iaz} \cdot CN_{ie,iel,iaz}$$

Here $CN_{ie,iel,iaz}$ is the "Count -> partial density" conversion factor

$$CN_{ie,iel,iaz} = \Delta W_{ie,iel,iaz}/(GV_{ie,iel,iaz} \cdot V^4_{ie,iel,iaz})$$

Thus to calculate moments you need just $CN_{ie,iel,iaz}$ and corresponding velocity components which are calculated as follows:

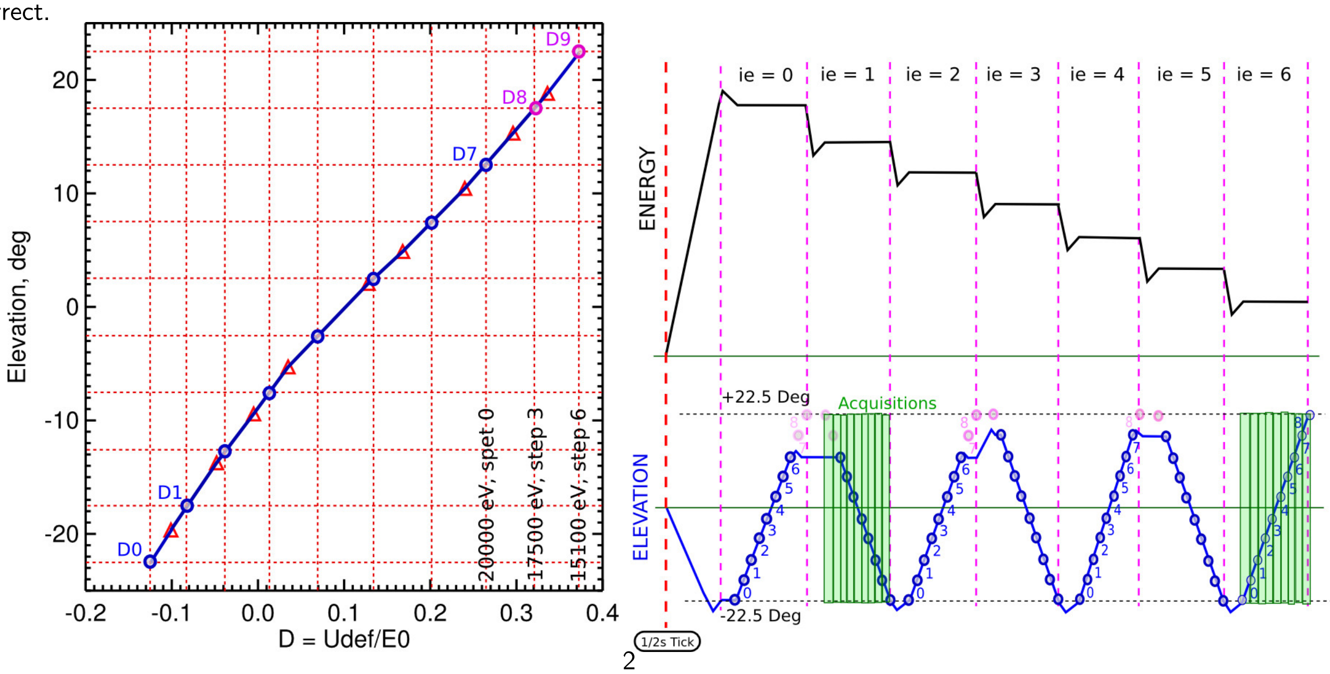
$$V^X_{ie,iel,iaz} = V_{ie,iel,iaz} \cdot cos(Az_{iaz}) \cdot cos(Elev_{ie,iel,iaz})$$

$$V^Y_{ie,iel,iaz} = V_{ie,iel,iaz} \cdot sin(Az_{iaz}) \cdot cos(Elev_{ie,iel,iaz})$$

$$V^Z_{ie,iel,iaz} = V_{ie,iel,iaz} \cdot sin(Elev_{ie,iel,iaz})$$

4. The PAS elevation sweep constraints

The figure below show the PAS calibration results and the PAS measurement scenario. The left panel shows the PAS calibration results with the elevation windows grid. The vertical axis is the elevation angle and the horizontal axis is the normalized deflector voltage. The blue and magenta circles show the nodes of the PAS deflector sweep. PAS accumulates data in the elevation window 0 when the deflector sweeps between the nodes D0 and D1. Because the high voltage power supply has a finite range, some nodes are unreachable for the high energies. The node 9 is reachable for $ie \geq 6$ and node 9 is reachable for $ie \geq 3$. Thus for the energy steps $ie < 6$ the sweeping procedure differs from the common one. The right panel of the figure shows the corresponding sweeping diagram. The top black line is the energy stepping, and the bottom blue line is the elevation sweeping profile. For, for instance, $ie = 0$ the elevation performs the sweep until the node D7 only (the last proper elevation window is #6). Then the Deflector voltage is constant until it start to going down for $ie = 1$ sweep. Note that PAS continues data acquisition even if the elevation window is not correct.



DPU shall remove from moments calculation the energy/elevation bins indicate by "X" in the table below:

ie/iel	0	1	5	......	6	7	8
0						X	X
1						X	X
2						X	X
3							X
4							X
5							X
6							
......							
95							

5. The onboard constants calculated from the full calibration parameters provided from the ground.
a. Proton velocity value wich depends on energy bin index ie only:

$$V_{ie} = 13.85e5 \cdot \sqrt{E_{ie}^{AV}}, cm/s$$

Here E_{ie}^{AV} is the $E_{ie,iel,iaz}$ averaged over Elevation and Azimuth
b. Elevation angle which depends on elevation bin number iel only:

$$Elev_{iel} = \sum_{ie,iaz} Elev_{ie,iel,iaz}/99.0$$

c. Sines and cosines of the main directions, calculated from Az_{iaz} and $Elev_{iel}$

$$SinAz_{iaz},\ CosAz_{iaz},\ SinElev_{iel},\ CosElev_{iel}$$

d. The "Count -> partial density conversion factor $CN_{ie,iel,iaz}$ (see section 3).

6. Onboard moments calculation and corresponding constants
Now we calculate N, V, and pressure tensor according to Vito's document.

a. Partial density:

$$\Delta n_{ie,iel,iaz} = Count_{ie,iel,iaz} \cdot CN_{ie,iel,iaz}$$

b. Number density:

$$n = \sum_{ie,iel,iaz} \Delta n_{ie,iel,iaz}$$

c. Velocity (an example for Vx):

$$V_X = (\sum_{ie,iel,iaz} V_{ie} \cdot CosAx_{iaz} \cdot CosElev_{iel} \cdot \Delta n_{ie,iel,iaz})/n$$

d. Pressure Tensor (an example for Pxy):

$$P_{XY}/m = \sum_{ie,iel,iaz} (V_{ie}^X - V_X)(V_{ie}^Y - V_Y)\Delta n_{ie,iel,iaz}$$

$$P_{XY}/m = \sum_{ie,iel,iaz} V_{ie}^X CosAx_{iaz} \cdot CosElev_{iel} \cdot V_{ie}^Y SinAx_{iaz} \cdot CosElev_{iel} \cdot \Delta n_{ie,iel,iaz} - n \cdot V_X \cdot V_Y$$

7. Which constant shall PAS provide

Thus PAS shall provide following arrays for the moment calculations:

Array	Size	Comment
$CN_{ie,iel,iaz}$	$96 \times 9 \times 11$	Density conversion factor
V_{ie}	96	Velocity, cm/s
$SinAz_{iaz}$	11	$sin(Az_{iaz})$
$CosAz_{iaz}$	11	$cos(Az_{iaz})$
$SinElev_{iel}$	9	$sin(Elev_{iel})$
$CosElev_{iel}$	9	$cos(Elev_{iel})$

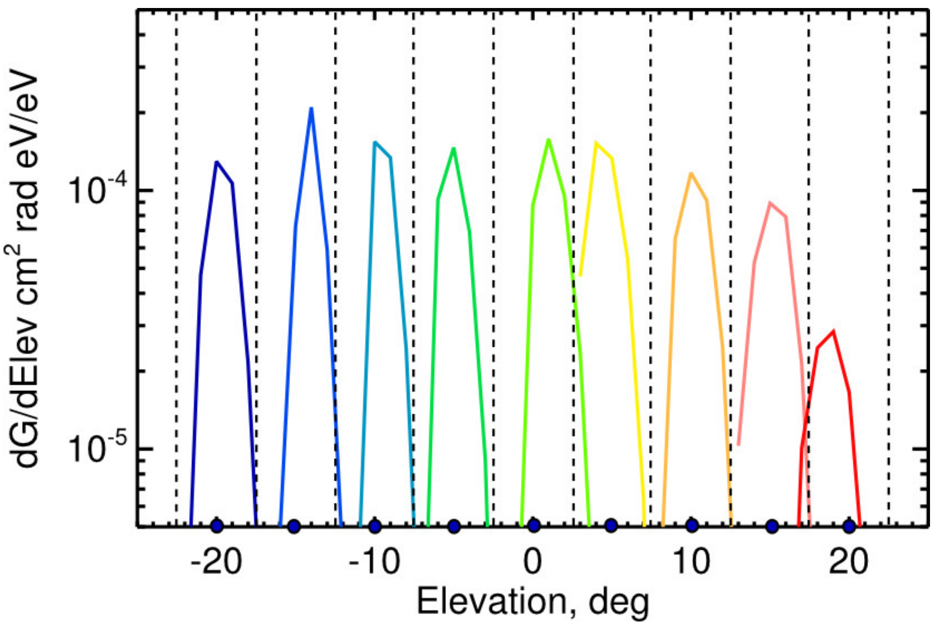
8. The output values and their precisions

The transmitting values are:

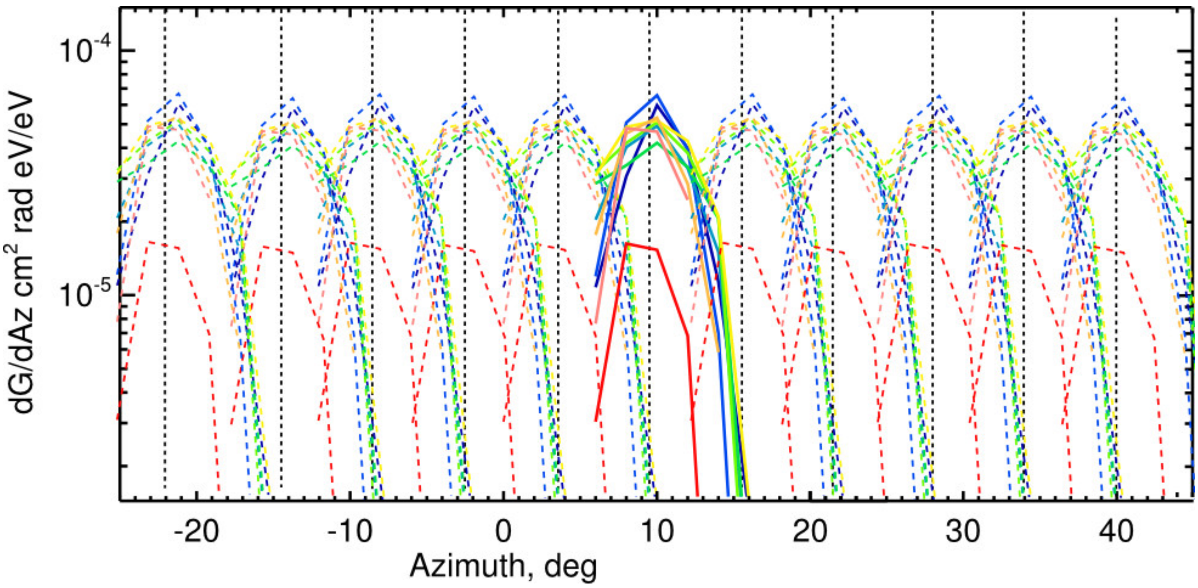
Moment	Precision	Comment
n	int16	Number density, scale and offset needed to decode
V_X	int16	Number density, scale and offset needed to decode
V_Y	int16	Number density, scale and offset needed to decode
V_Z	int16	Number density, scale and offset needed to decode
P_{XX}/m	float16	Pressure tensor components
P_{YY}/m	float16	
P_{ZZ}/m	float16	
P_{XY}/m	float16	
P_{XZ}/m	float16	
P_{YZ}/m	float16	

9. EQM calibration data and corresponding "almost flight" calibration tables:

The calibrated angular responses expressed in the geometrical factor are shown in the following two figures:



Elevation response for CEM #5. The vertical dashed lines correspond to the boundaries of the azimuthal bins.



Azimuth response. The measured curves for CEM #5 are expanded to the entire azimuthal range. Different colors correspond to the different elevation angles, as well as in the previous figure.

Since only one detector had the flight-like configuration, we have extrapolated calibration data to all detectors. Note the the elevation responses are just "instance snapshots" of the PAS angular respopnse continuously sweeping over the elevation bins (dashed black lines).

On the base of this data, the calibartion tables (see section 7) have been preparated and saved in the files as follows:

Table	File	Comment
$CN_{ie,iel,iaz}$	PAS_EQM_CN_array_14_Feb_2017.txt	
V_{ie}	PAS_EQM_V_array_14_Feb_2017.txt	
$Elev_{iel}$	PAS_EQM_El_array_14_Feb_2017.txt	$SinElev_{iel}$ and $CosElev_{iel}$
Az_{iaz}	PAS_EQM_Az_array_14_Feb_2017.txt	$SinAz_{iaz}$ and $CosAz_{iaz}$

All files are self-descibed.

The PAS PFM calibration record will be delivered in the same format.