Title

PLATO FEE-to-DPU Interface Requirement Document (IRD)

Subtitle Issue for Unit PDRs

Ref. **PLATO-DLR-PL-ICD-0002**

Issue **1.2**

Date 2018/12/14

---	Name	Date	Signature
Prepared:	Karsten Westerdorff, DLR		
Checked:	Class Ziemke, DLR		
---	Bernd Ulmer, DLR/IBU		
---	Alexander Koncz, DLR		
---	Philippe Plasson, LESIA		
---	Tom Kennedy, MSSL		
---	Jörg Würker, OHB		
Approved:	Carolin Wimmer, OHB		
Authorized:	Gisbert Peter, DLR		

CHANGE HISTORY

Issue	Change	Approved	Date
0.1	**Initial release**	K. Westerdorff	2017/04/05
0.2	Changed or added FEE-DPU-IF-531,-532, -533, -914, -927, -539, -912, -913, -543, -925, -926, -548, -923, -924, -937, -900, -551, -921, -919, -920, -556, -558, -559, -562, -563, -565, -566, -567, -934, -587, -589, -828, -830, -840, -844, -936, -863, -935, -594, -871, -873, -874, -875, -878, -881, -888, -891, -896, -897	K. Westerdorff	2017/10/23
0.3	Changed FEE-DPU-IF-541, 544, 925, 926, 923, 924, 551, 919, 920, 554, 556, 558, 561, 562, 563, 565, 927, 543, 546, 925, 923, 900, 905, 920, 561, 570, 573, 574, 575, 578, 579, 580, 581, 582, 583, 584, 585, 586, 588, 589, 591, 592, 826, 827, 828, 832, 833, 834, 835, 836, 837, 838, 839, 840, 842, 844, 845, 936, 846, 847, 848, 849, 850, 852, 853, 855, 856, 858, 859, 860, 861, 862, 863, 864, 865, 873, 874, 878, 879, 881, 896, 897, 895 Deleted FEE-DPU-IF-587, 590, 829, 830, 868, 841, 907, 843, 854, 935 Added FEE-DPU-IF-964, 953, 954, 941, 943, 949, 966, 967	K. Westerdorff	2018/05/30
1.0	Changed, added or deleted FEE-DPU-IF-529, 530, 531, 532, 533, 914, 539, 543, 544, 970, 545, 972, 546, 925, 547, 926, 548, 923, 969, 905, 968, 924, 974, 967	K. Westerdorff	2018/09/25
1.1	Rename of Doc-ID from IC-0002 to ICD-0002 as outcome of the PL-PDR (action item 38010)	G. Peter	2018/11/21
1.2	added verification methods	K. Westerdorff	2018/12/14

EXPORTED MODULES FROM IBM DOORS

Module				**Module ID**
/PLATO/Payload/Software/FEE-DPU IRD				**0000036e**
Exported Version	Current Version, Last change: 14.12.2018 doors://RMC-075001WTS.intra.dlr.de:36677/?version=2&prodID=0&urn=urn:telelogic::1-53ede90401644d27-M-0000036e			
Exported View	Export View			
Pages in this document	5 - 42			
Changes	Baseline	Created On	Created by	Description
	1.0	24.09.2018	west_ka	first official issue
	1.1	14.12.2018	west_ka	
	1.2	14.12.2018	west_ka	

TABLE OF CONTENTS

1 PRESENTATION OF THE DOCUMENT

1.1 Purpose of the Document

This document describes the requirements for the interface between the F-FEE / F-DPU and N-FEE / N-DPU. Because both interfaces are equal in most ways, the requirements are summarized in a common document. Spedific requirements for F-FEE / F-DPU or N-FEE / N-DPU interface will be highlighted.
Because the FEE must support a subset of RMAP only, this documents tailors the according standard. Interface-relevant details of the FEE- and DPU-implementation will be specified as well. Finally the FDIR measures will be covered by this document.

1.2 Application

This document shall be used as an applicable document for the PLATO sub-units N-FEE, N-DPU, F-FEE and F-DPU development.

With reference to the upper level requirements, this ICD specifies all network layers of the corresponding sub-units.

1.3 Responsibility

This document is prepared and written by DLR. DLR will update the document with the input of all parties of the concerning sub-systems.

2 REFERENCES

2.1 Applicable Documents

	Title	Reference
AD01		
AD02		
AD03		
AD04		
AD05		

2.2 Applicable ECSS Standards

	Title	Reference
AD20	SpaceWire - Links, nodes, routers and networks	ECSS-E-ST-50-12C (31/07/2008)
AD21	SpaceWire protocol identification	ECSS-E-ST-50-51C (5 February 2010)
AD22	SpaceWire - Remote memory access protocol	ECSS-E-ST-50-52C (5 February 2010)

Remark: At next issue, ECSS-E-ST-50-12C might be renumbered to: ECSS-E-ST-50-50E

2.3 Reference Documents

	Title	Reference
RD01	PLATO FEE Windowing - Technical Note	PLATO-DLR-PL-TN-018, Issue 1.3 (11/2016)
RD02	PLATO SIMICAm Patter Requirements	PLATO-LESIA-PL-TN-023, Issue 1.1 (03/2017)
RD03	PLATO CCD Definition	PLATO-MSSL-PL-TN-008

2.4 Glossary & Acronyms

AIT	Assembly, Integration and Test
AIV	Assembly, Integration and Verification
AOCS	Attitude and Orbit Control System
ASW	Application SoftWare
BSW	Boot SoftWare
CCD	Charge Coupled Device
CIDL	Configuration Item Data Lists
CNES	Centre National d'Études Spatiales
DLR	German Aerospace Center
DMA	Direct Memory Access
DPS	Data Processing System
DPU	Data Processing Unit
DSU	Debug Support Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EGSE	Electrical Ground Support Equipment
EM	Engineering Model

ESA	European Space Agency
ESTEC	European Space Research & Technology Centre
F-DPU	Fast camera DPU
FEE	Front End Electronics
FEU	Fast Electronics Unit
FGS	Fine Guidance System
FM	Flight Model
FoV	Field of View
FPA	Focal Plane Assembly
Gb	Gigabit
GS	Ground Station
GSE	Ground Support Equipment
HK	Housekeeping data
HKTM	Housekeeping telemetry
HW	Hardware
I/F	Interface
ICU	Instrument Control Unit
kbps	Kilobit per second
Mb	Megabit
Mbps	Megabit per second
Mpx	Mega-pixel
MEU	Main Electronics Unit
MGSE	Mechanical Ground Support Equipment
MOC	Mission Operation Centre
N-DPU	Normal camera DPU
OB	Optical bench
OBCP	On-Board Control Procedure
OGSE	Optical Ground Support Equipment
P/L	Payload
PDAAS	Plato Data Acquisition and Analysis System
PDC	PLATO ground Data Centre
PFM	Proto Flight Models
PI	Principal Investigator
PICD	Payload Interface Control Document (Part B)
PLATO	PLAnetary Transits and Oscillations
PLM	Payload Module
PLTM	Payload Telemetry
PPLC	PLATO Payload Consortium
ppm	part per million
Px	Pixel
QM	Qualification Model
RMAP	Remote Memory Access Protocol
SOC	Science Operation Centre
SpW	SpaceWire
STM	Structural Thermal Model
SVM	Service Module
SW	Software
SWT	Science Working Team
TBC	To Be Confirmed
TBD	To Be Determined/Defined
TC	Telecommand
TM	Telemetry
CCSDS	Consultative Committee for Space Data Systems
PUS	Packet Utilization Standard

ECSS	European Cooperation for Space Standardization
APID	Application Identifier
PID	Process Identifier
PCAT	Packet Category
SICD	Software Interface Control Document

3 MISSION AND BACKGROUND

3.1 The PLATO Mission

PLATO is an M-class mission candidate of the European Space Agency's Science programme Cosmic Vision 2015-2025 foreseen to be launched by 2026. "PLAnetary Transits and Oscillations of stars" aims to characterise exoplanetary systems by detecting planetary transits and conducting asteroseismology of their parent stars.

PLATO is the next generation planetary transit experiment; its objective is to characterize exoplanets and their host stars in the solar neighbourhood. While it builds on the heritage from CoRoT and Kepler, the major breakthrough to be achieved by PLATO will come from its strong focus on bright targets, typically with mV≤11. The PLATO targets will also include a large number of very bright and nearby stars, with mV≤8.

The prime science goals of PLATO are:

* the detection and characterization of exoplanetary systems of all kinds, including both the planets and their host stars, reaching down to small, terrestrial planets in the habitable zone;
* the identification of suitable targets for future, more detailed characterization, including a spectroscopic search for biomarkers in nearby habitable exoplanets;
* a full characterisation of the planet host stars, via asteroseismic analysis: this will provide us with the masses, radii and ages of the host stars, from which masses, radii and ages of the detected planets will be determined.

These ambitious goals will be reached by ultra-high precision, long (few years), uninterrupted photometric monitoring in the visible of very large samples of bright stars, which can only be done from space. The resulting high quality light curves will be used on the one hand to detect planetary transits, as well as to measure their characteristics, and on the other hand to provide a seismic analysis of the host stars of the detected planets, from which precise measurements of their radii, masses, and ages will be derived. For the brightest targets, planets are also expected to be detectable through the modulation of stellar light reflected on the planet surface, and/or through the astrometric wobble induced on the star by the planet orbital motion.

The PLATO space-based data will be complemented by ground-based follow-up observations, in particular very precise radial velocity monitoring, which will be used to confirm the planetary nature of the detected events and to measure the planet masses.

The full set of parameters of the systems with detected exoplanets will thus be measured, including all characteristics of the host stars and their orbits, radii, masses, and ages of the planets. Measurements of the radii and masses will be used to derive the planet mean densities and therefore will give insight on their internal structure and composition. The orbital parameters, together with the precise knowledge of all characteristics of the host star, will enable us to estimate the temperature and radiation environment of the planets. Finally, the knowledge of the age of the exoplanetary systems will allow us to put them in an evolutionary perspective.

See **[RD1]** for further details on the PLATO mission.

3.2 Instrument Architecture

The instrumental concept proposed by the PLATO Payload Consortium is based on a multi-camera approach, involving a set of several normal instruments monitoring stars fainter than mV=8, plus a low number of fast instruments observing extremely bright stars with magnitudes brighter than mV=8.

The telescope is based on a fully dioptric design, working in an extended visible light range. It has been designed to be able to observe a very large field, with respect to a sufficient pupil diameter.

The 24 normal cameras are arranged in four sub-groups of 6 cameras. All 6 cameras of each sub-group have exactly the same Field of View (FOV), and the lines of sight of the four sub-groups are offset by +/-9,2° of their FOV of about 38°. This particular configuration allows surveying a very large field at each pointing, with various parts of the field monitored by 24, 12 or 6 normal cameras.

This strategy optimizes both the number of targets observed at a given noise level and their brightness. It is assumed that the satellite will be rotated around the mean line of sight by 90° every 3 months, resulting in a continuous survey of exactly the same region of the sky.

Each camera is equipped with its own CCD focal plane array, comprised of 4 CCDs. The CCDs work in full frame mode for the normal cameras, and in frame transfer mode for the fast cameras.

Each FPA is associated to a Front End Electronics (FEE). The camera (after Instruments tests) is delivered for PLM AIT as one unit. The camera is delivered with FEE and FPA connected together by their flexi-cables. For safety reasons, these links shall never be disconnected after the delivery of the camera to PLM.

There are several units, the AEUs, which provide secondary voltages for the FEEs. 2 N-AEU boxes provides voltages for the normal FEEs/cameras, one N-AEU for one batch of 12 normal cameras. One F-AEU provides the voltages for the two fast FEEs/cameras. Additionally the F-AEU contains a synchronization module which provides hardware synchronization signals to the FEEs (synchronizing the CCD read-out), power supplies (synchronizing the DC/DC converters) in the AEU and to the SVM (synchronizing the thermal temperature control of the TOUs).

3.3 Data Processing System (DPS) Architecture

The PLATO payload data processing system is made up of the DPUs and the ICU, with data routed through a SpaceWire network. The ICUs are connected to the SVM through SpaceWire links.

There are 12 normal DPUs. Each N-DPU is responsible for processing the data of 2 normal cameras. The processing cadence for N-DPUs is 25 sec.

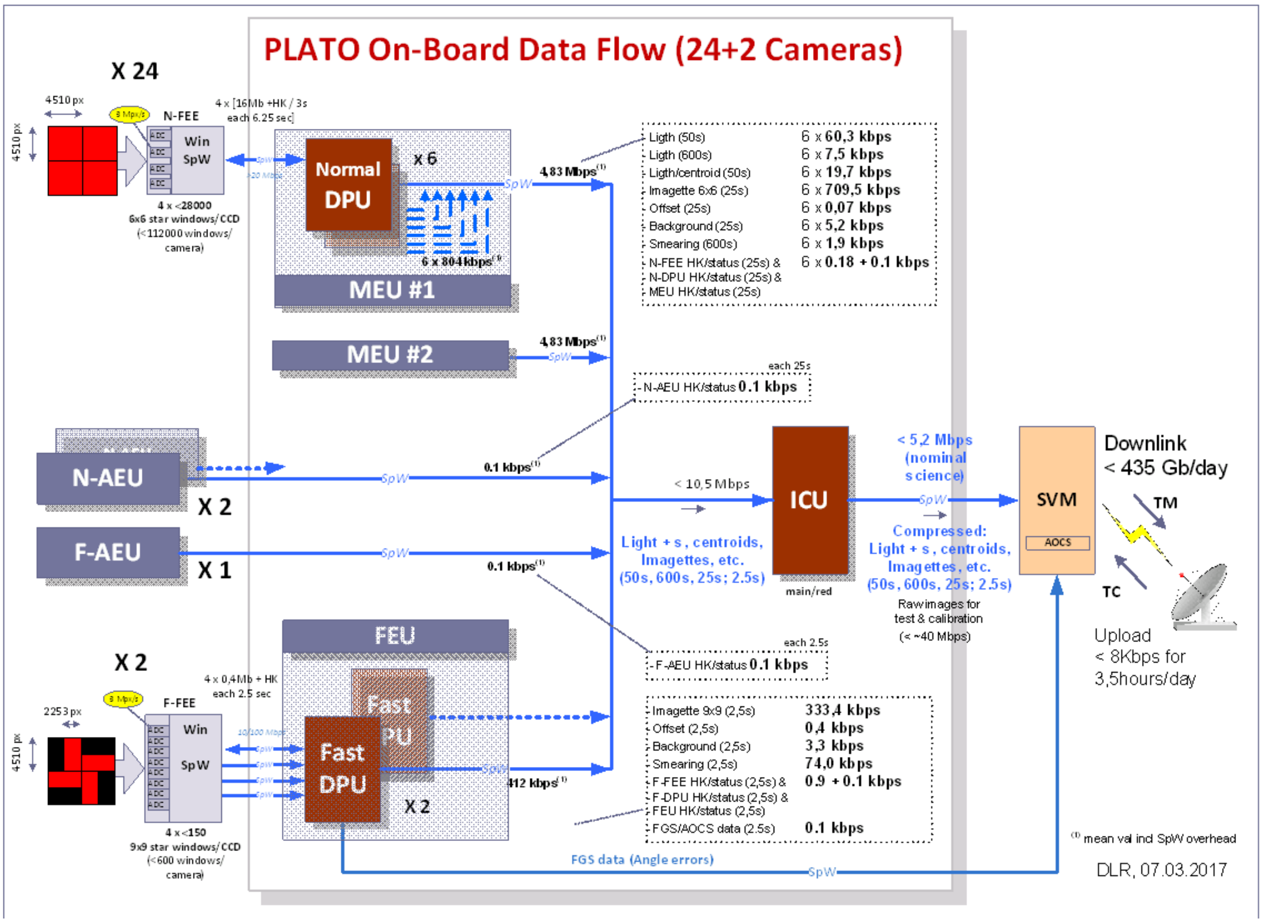
There are 2 fast DPUs gathered in one electronic box named FEU (Fast Electronic Unit). Each F-DPU is responsible for processing the data of one fast camera. The processing cadence for F-DPUs is 2.5 sec.

The F-DPUs have a supplementary function: they are responsible for providing angle error data as Fine Guidance System (FGS) measurements directly to the SVM AOCS.

There are 2 ICU channels which work in cold redundancy. The ICU is responsible for the management of the payload, the communication with the Service Module (SVM), the compression of scientific data before transmitting them as telemetry to the SVM.

The following figure gives an overview of the PLATO data processing system architecture and of the data flow rates. It focuses on the sharing of the main functions and the data flows. It is a simplified view of the hardware architecture.

Figure: PLATO on-board data flow

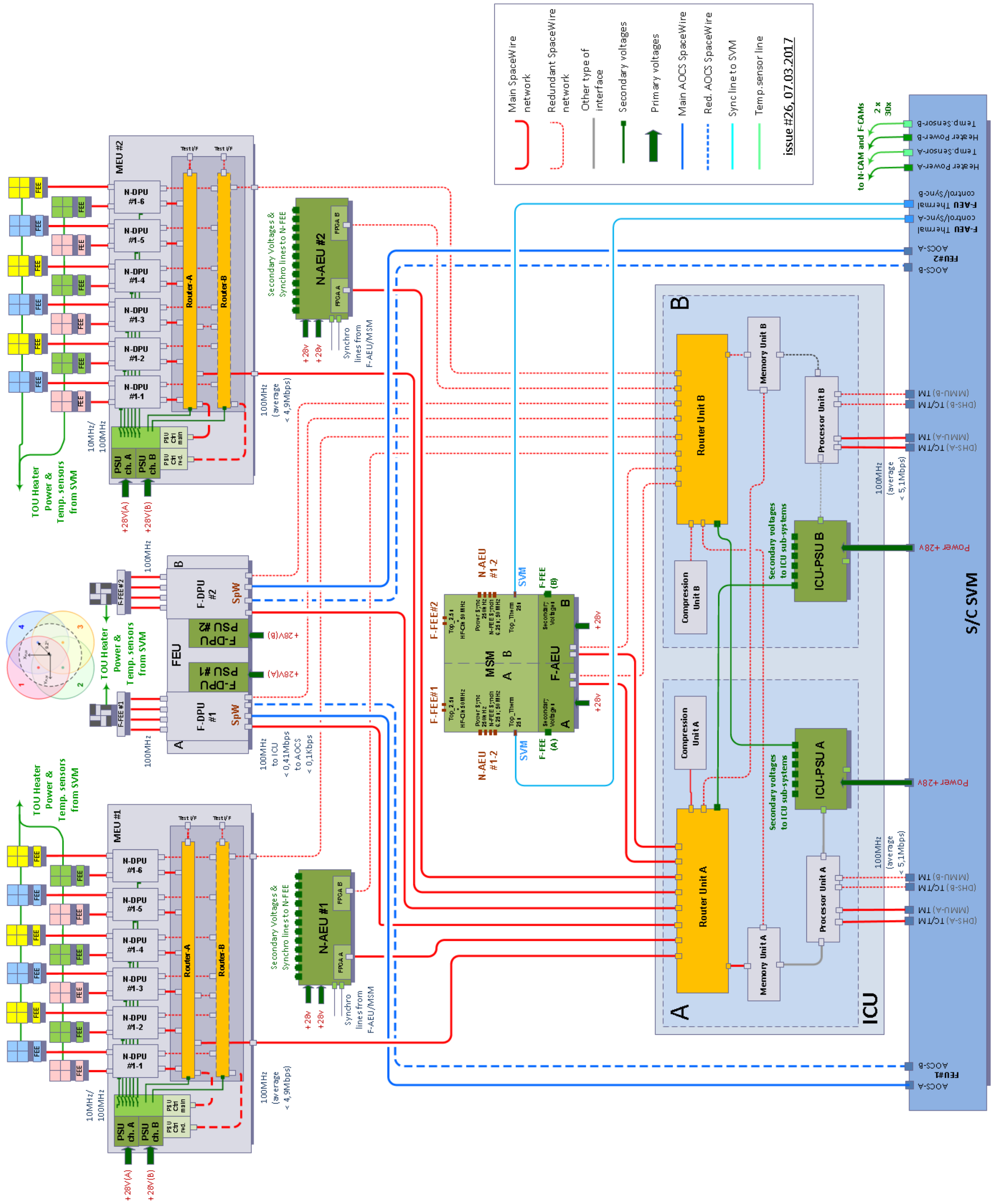


Due to fault tolerance reasons and in order to optimize the resources (mass, volume, harness), the physical implementation of the architecture described above foresees to split the 12 N-DPUs in 2 groups of 6 N-DPUs. Each group of 6 N-DPUs is gathered in a box called Main Electronic Unit (MEU).

In the same way, the two cold redundant ICU channels are gathered in a same box.

The figure below shows the Payload architecture:

Figure: PLATO Payload Electrical Architecture



4 Spacewire Interface

Between FEE and DPU Spacewire is the only electrical interface. RMAP and a PLATO specific data-protocol will be used at the higher layers of the Spacewire-network.

The physical-layer, character-layer, exchange-layer, packet-layer and the network-layer of the Spacewire interface are defined in the URDs.

	Title:	**Spacewire Error Register**
	Justif.:	AD20 chapter 5.5.7
	Verif.:	Review-of-Design
FEE-DPU-IF-927	The FEE shall provide a Spacewire error register to store the reason of a Spacewire disconnect. The reason for a disconnect could be disconnect by DPU, parity error, escape error, character sequence error and credit-error.	

	Title:	**DPU Spacewire-Address**
	Verif.:	Review-of-Design
FEE-DPU-IF-534	All Spacewire packets sent by the FEE and targeting the DPU shall have the logical address 0x50.	

Note: This address is only valid inside the FEE-DPU network, but will not be visible in other Spacewire-networks in the payload-system.

	Title:	**FEE Spacewire-Address**
	Verif.:	Review-of-Design
FEE-DPU-IF-536	All Spacewire packets sent by the DPU and targeting the FEE shall have the logical address 0x51.	

Note: This address is only valid inside the FEE-DPU network, but will not be visible in other Spacewire-networks in the payload-system. Therefore, the FEE cannot be directly addressed from the platform or the ICU.

	Title:	**Spacewire Routing**
	Verif.:	Review-of-Design
FEE-DPU-IF-538	The Spacewire-connection between FEE and DPU shall be direct, without any router. Logical address routing is used for the FEE-DPU interface.	

	Title:	**Spacewire Timecode Generation**
	Verif.:	Test
FEE-DPU-IF-539	In all modes, the FEE shall generate a time-code on arrival of the external synchronization signal from the AEU.	

Note: The time-code value shall be compliant to [AD22]. The lower 6-bit of the time-code shall be incremented on every synchronization-signal. The control-flags (bit 7 and 8) shall be set to 0.

Note: The time-code will signal the beginning of the read-out phase.

Note: The synchronization signal is generated by the F-AEU and will be provided as long as the FEE is powered.

	Title:	**Timecode Link (F-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-541	The Spacewire time-code of the F-FEE shall be send only over one Spacewire-link. The Spacewire-link shall be selectable in the F-FEE-configuration.	

	Title:	**Timecode Accuracy**
	Verif.:	Test
FEE-DPU-IF-913	The time-code shall be sent within 1 microsecond after arrival of the external synchronization signal.	

5 FEE-Modes

The FEE control interface is based on modes. These modes narrow the flexibility, but keep the commanding by the DPU in a well-defined way. This will simplify the commanding of the FEEs and will protect the CCD and electronics.
For operating the FEE the standard sequence is:
1. The DPU checks if the FEE is in the expected mode.
2. The DPU checks the status und sets/changes parameters.
3. The DPU requests a mode-change, that usually will be executed by the FEE on reception of the external sync-impuls.
4. The DPU checks if the mode-change was successful.
Nevertheless, the mode-approach leaves some room for flexible operation inside the modes. For instance: If a complex power-on sequence is used, the enable commands and the checks can be done "manually" by the DPU before requesting the change into the stand-by mode.

	Title:	**FEE Modes**
	Verif.:	Review-of-Design
FEE-DPU-IF-543	The FEE shall support the following modes: - FEE_ON = On Mode - FEE_STANDBY = Stand-By Mode - FEE_FULLIMAGE = Full-Image-Mode - FEE_TEST_FULLIMAGE_PATTERN = Full-Image-Pattern-Mode - FEE_WINDOWING = Windowing-Mode - FEE_TEST_WINDOWING_PATTERN = Windowing-Pattern-Mode - FEE_TEST_PARTIAL_READOUT = Partial-Readout-Mode	

Note: Additional modes might be defined by FEE teams.

	Title:	**Default Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-544	After power-on the FEE shall enter the on-mode before initializing the Spacewire interface.	

	Title:	**On-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-970	The FEE shall be in on-mode, if the CCD and the FEE subsystems are not powered.	

	Title:	**Stand-By-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-545	The FEE shall be in stand-by-mode, if the CCD and the FEE subsystems are powered, but no image-data is delivered to the DPU.	

Note: The configuration of the FEE should be done on-mode and in stand-by-mode. Most of the configuration-settings should be locked outside the these modes.

	Title:	**Mode Entering**
	Verif.:	Review-of-Design
FEE-DPU-IF-972	All modes shall be entered only from stand-by-mode.	

	Title:	**Full-Image-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-546	The FEE shall be in full-image-mode, if one half-CCD (N-FEE) or four half-CCDs (F-FEE) are transmitted to the DPU.	

Note: Because the bandwidth to the DPU is limited, a complete CCD-image cannot be transmitted at once. Therefore the transmission is stretched over two read-out cycles.

	Title:	**Full-Image-Pattern-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-925	The FEE shall be in full-image-pattern mode, if the FEE sends pattern instead of video-data to the DPU, but beside this behaves like in full-image mode.	

	Title:	**Windowing-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-547	The FEE shall be in windowing-mode, if the full CCD will be read out, but only regions of interests will be transmitted to the DPU.	

	Title:	**Windowing-Pattern-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-926	The FEE shall be in windowing-pattern mode, if the FEE sends pattern instead of video-data, but beside this behaves like in windowing mode.	

	Title:	**Partial-Readout-Mode**
	Verif.:	Review-of-Design
FEE-DPU-IF-548	The FEE shall be in partial readout mode, if most of the lines are dumped during readout to avoid over-saturation in ambient conditions.	

	Title:	**RMAP Mode Change Request**
	Verif.:	Review-of-Design
FEE-DPU-IF-974	The FEE shall accept mode-change request as a single RMAP write-request.	

Note: A mode-change can be done by the FEE autonomously as FDIR measure.

	Title:	**Mode Change Synchronization**
	Verif.:	Test
FEE-DPU-IF-923	The FEE shall change the mode at the next 25 second sync-signal at the N-FEE and at the next 2.5 second sync-signal in the F-FEE, except the change between on-mode and standby-mode	

Note: The FEE might generate the sync-signal internally.

	Title:	**Immediate Stand-By**
	Verif.:	Test
FEE-DPU-IF-924	The FEE shall return to stand-by-mode immediately after the special command "immediate stand-by" was received from the DPU.	

	Title:	**Mode Status in HK**
	Verif.:	Review-of-Design
FEE-DPU-IF-969	The current FEE-mode shall be visible in a HK-register and in the HK data-packet.	

	Title:	**CCD Read Out Order (N-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-937	The CCD sequence for the read-out shall be configurable in the N-FEE. I.e. all CCDs will be read in a 25 second cycle, but the order can be defined by the N-DPU.	

	Title:	**Constant CCD Read-Out (N-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-964	The N-FEE shall provide the option for reading always the same CCD (no CCD sequencing). The CCD, that is constantly read, shall be selectable by the N-DPU.	

6 Windowing

	Title:	**Coordinate System and Orientation**
	Verif.:	Review-of-Design
FEE-DPU-IF-900	The coordinate system for the windowing is derived from the CCD read-out scheme. Right and left side of the CCD have seperate coordinate-systems. The origin of both coordinate-systems is the first pixel read from the CCD.	

Figure: Coordinate-system of the N-camera and the orientation of the windows

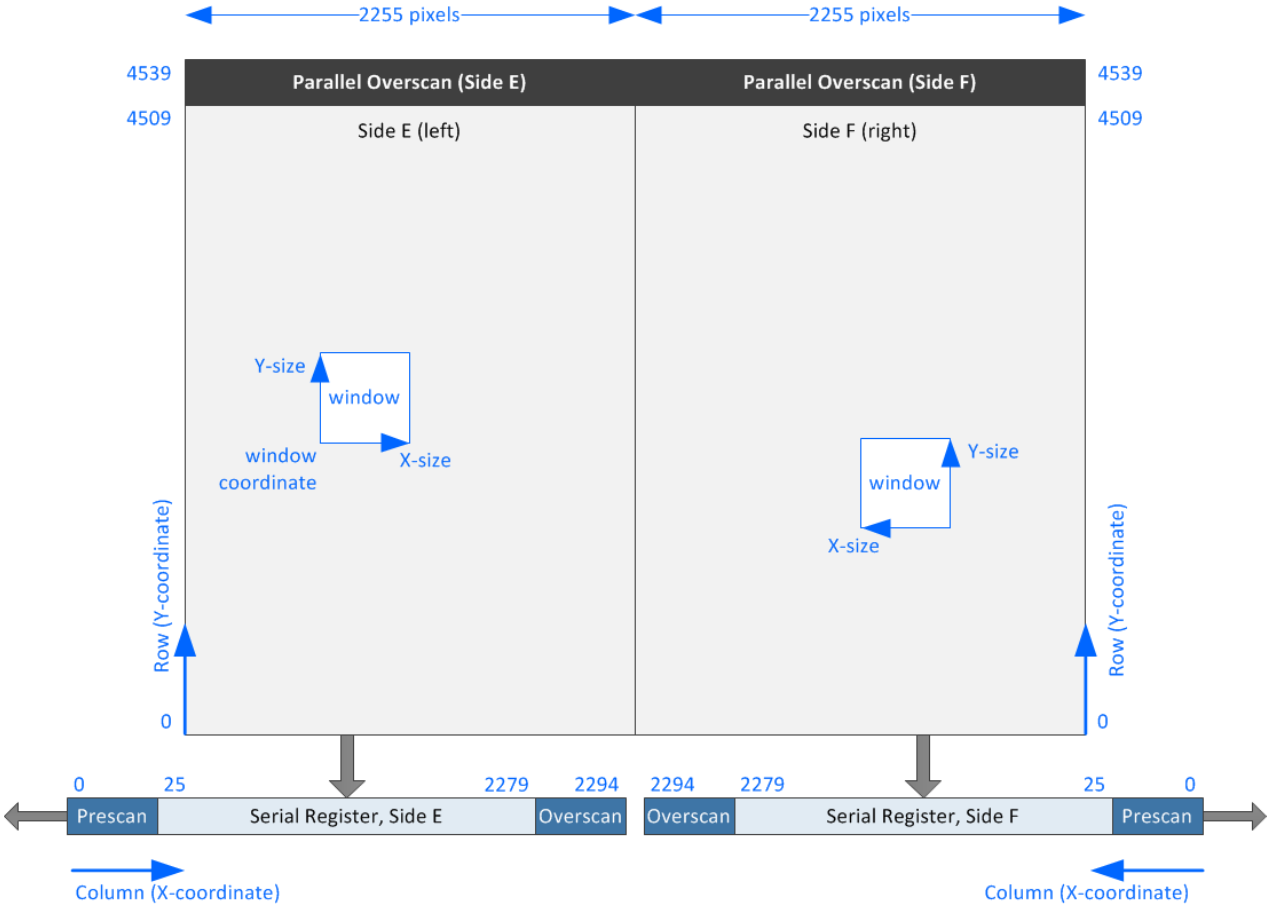
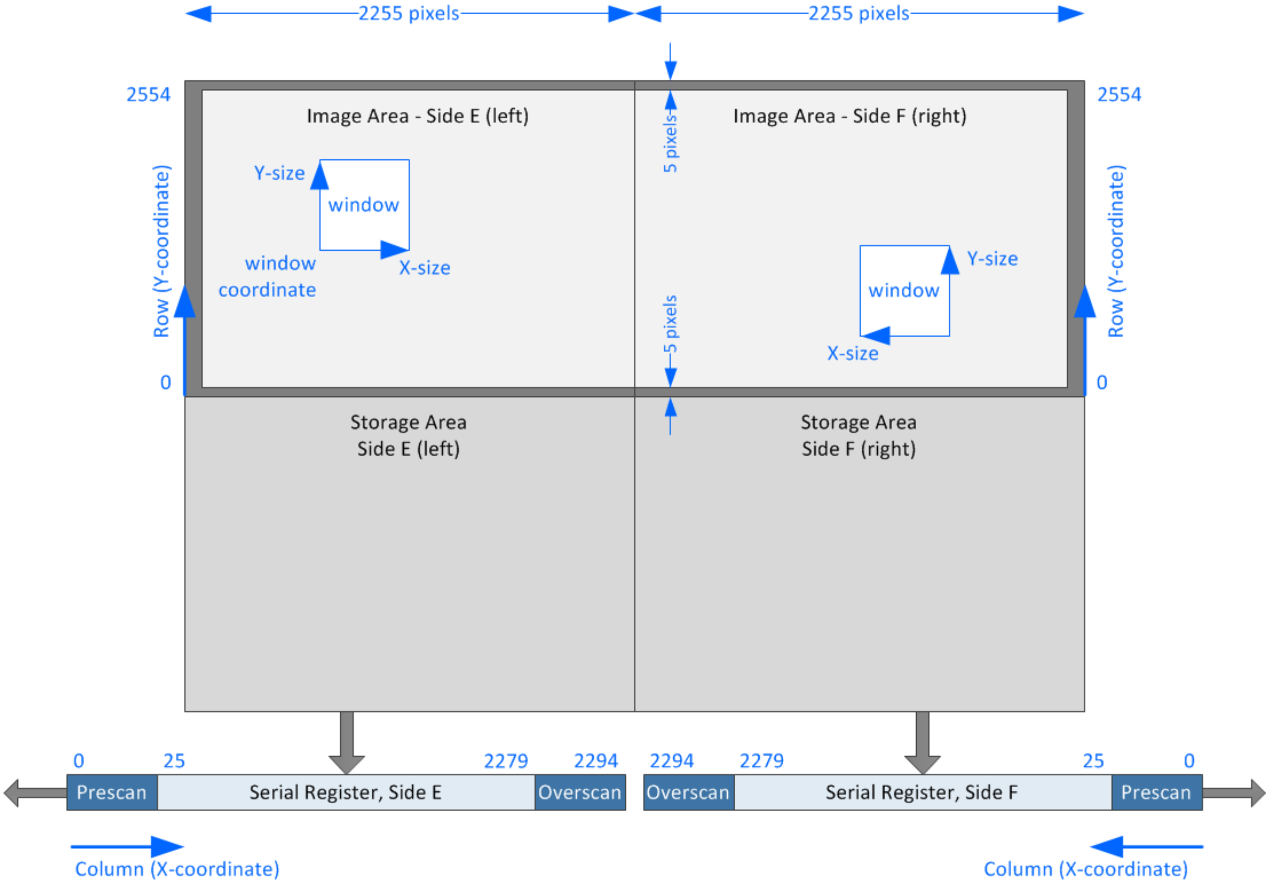
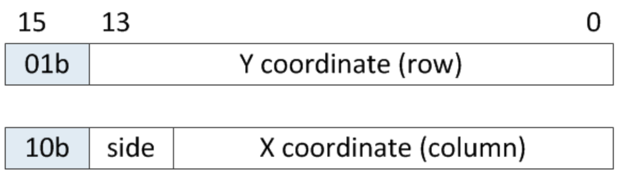


Figure: Coordinate-system of the F-camera and the orientation of the windows



	Title:	**Window-Parameters**
	Verif.:	Review-of-Design
FEE-DPU-IF-551	A window shall be defined by the following three parameters: - Y-coordinate == CCD row - X-coordinate == CCD column (bits 12:0) - CCD-side (bit 13: 0 = left, 1 = right) Each parameter is 16-bit width, containing a 2-bit identifier and a 14-bit value.	

Figure: Structure of the window parameters



Note: The active window list is an array generated in the N-FEE, which holds all window touched by the current read-out line. The example in requirement FEE-DPU-IF-559 highlights the active windows in red.

	Title:	**Window Lists**
	Verif.:	Review-of-Design
FEE-DPU-IF-553	A window shall be defined for a specific CCD. So, there shall be four separate lists with window-definitions.	

	Title:	**Window List Length (N-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-905	The N-FEE shall store up to 150,000 window-coordinates, summarized over the four window-lists.	

Note: The number of window-coordinates consists of 112 thousend target windows + 30% for larger PSF, 3000 background windows and 1000 windows for saturated stars. The 30% more windows are required for targets, where the position on the CCD results in a larger PSF. Here the 6x6 pixel window is not sufficient and will be extended by additional windows. Alternatively the window-size could be increased. But this would result in a higher processing-load of the DPU.

	Title:	**Window List Length (F-FEE)**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-906	The F-FEE shall store up to 700 window-coordinates, summarized over the four window-lists.	

	Title:	**Windows per CCD (F-FEE)**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-921	The F-FEE shall be able to process 512 windows per CCD.	

	Title:	**Maximum Windows per Row (N-FEE)**
	Justif.:	Worst case estimation with 6 pixel wide windows cover all pixels of the line without overlapping = (2255 pixels per line + 40 pre-/overscan) / 6 * 2
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-919	The N-FEE shall be able to handle up to 766 active windows per line.	

	Title:	**Maximum Windows per Column (F-FEE)**
	Justif.:	Worst-case estimation for having a quarter of the windows in one line.
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-920	The F-FEE shall be able to handle up to 128 windows per column.	

	Title:	**Window-List Pointer Registers**
	Verif.:	Review-of-Design
FEE-DPU-IF-554	For each window-list the FEE shall contain two registers, which hold the address-pointer and the length of the list. The length shall be given in number of parameter-words (16-bit words). Pointer- and length-register shall be programmable by the DPU.	

Note: Because 4 window-lists must be handled, there shall be 4 pointer- and 4 length-registers.

	Title:	**Window Definition (N-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-556	A window list for the N-FEE shall have the following structure: The Y-coordinate shall be defined once per row. After the Y-coordinate tuples of X-coordinate follow. At a new row the next Y-coordinate is stated in the list.	

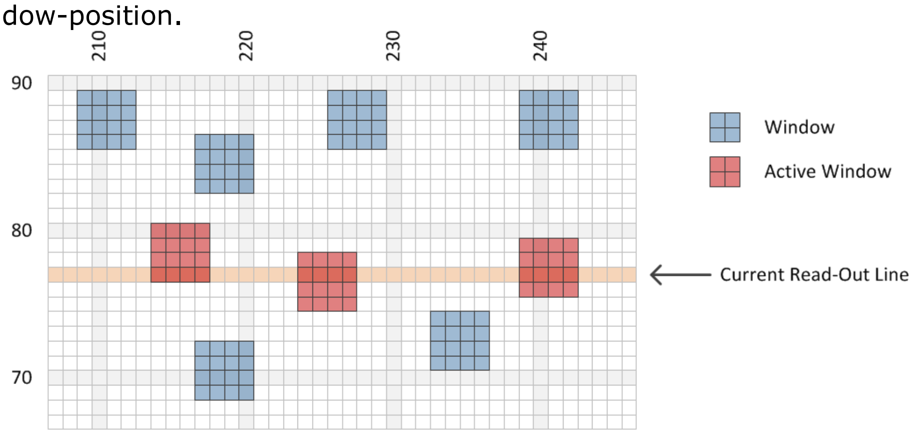
	Title:	**Window-List Sorting (N-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-558	The N-DPU shall upload window-list to the N-FEE, that shall be sorted first by Y-coordinate (row) and by second by X-coordinate (column).	

	Title:	**Window Definition (F-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-559	A window list for the F-FEE shall consist of X/Y-coordinate tuples.	

	Title:	**Window-List Sorting (F-FEE)**
	Verif.:	Review-of-Design
FEE-DPU-IF-561	The F-DPU shall upload window-list to the F-FEE, that shall be sorted first by X-coordinate (column) and second by Y-coordinate (row).	

Note: The sorting is different for N-FEE and F-FEE is different because of the amount of windows to handle and the resulting implementation. The F-FEE can scan all window-definition during a line read-out. In contrast the N-FEE must focus on the windows touching the current read-out line for preparing the bit-mask.

Figure: Window-list examples for N-FEE and F-FEE
For the N-FEE option A is without active-window-position (recommend) and option B is with active-window-position.



N-FEE
4045h (Y: 4000h + 69d)
80D9h (X: 8000h + 217d)
4047h (Y: 4000h + 71d)
80E9h (X: 8000h + 233d)
404Bh (Y: 4000h + 75d)
80E0h (X: 8000h + 224d)
404Ch (Y: 4000h + 76d)
80EFh (X: 8000h + 239d)
404Dh (Y: 4000h + 77d)
80D6h (X: 8000h + 214d)
4053h (Y: 4000h + 83d)
80D9h (X: 8000h + 217d)
4056h (Y: 4000h + 86d)
80D1h (X: 8000h + 209d)
80E2h (X: 8000h + 226d)
80EFh (X: 8000h + 239d)

F-FEE
80D1h (X: 8000h + 209d)
4056h (Y: 4000h + 86d)
80D6h (X: 8000h + 214d)
404Dh (Y: 4000h + 77d)
80D9h (X: 8000h + 217d)
4045h (Y: 4000h + 69d)
80D9h (X: 8000h + 217d)
4053h (Y: 4000h + 83d)
80E0h (X: 8000h + 224d)
404Bh (Y: 4000h + 75d)
80E2h (X: 8000h + 226d)
4056h (Y: 4000h + 86d)
80E9h (X: 8000h + 233d)
4047h (Y: 4000h + 71d)
80EFh (X: 8000h + 239d)
404Ch (Y: 4000h + 76d)
80EFh (X: 8000h + 239d)
4056h (Y: 4000h + 86d)

	Title:	**Window List Upload**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-562	The FEE shall be able to receive new window-lists outside the read-out phase.	

Note: At the F-FEE this could be done during shift to the CCD-store-section and after read-out. At the N-FEE new window coordinates could be uploaded during the read-out of another CCD.

	Title:	**Window Size**
	Verif.:	Review-of-Design
FEE-DPU-IF-563	All windows shall have the same size. The size of the windows shall be configurable in a range of 2x2 to 32x32 pixels in one-pixel steps. X and Y size can be different. During read-out the size shall be fixed.	

Note: The default imagette size for normal and fast cameras is 6x6 pixels. For the FGS a windows size of 9x9 will be needed from the F-FEE. At the first cycle, before the fine-pointing is established, bigger windows will be requested for the guide-stars. In this case the windows size can be increased to 32x32 pixels and only the 30 guide-stars will be transferred to the F-DPU.

	Title:	**Window Size Modification**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-565	The FEE shall provide the possibility to change the window-size outside read-out in windowing-mode.	

	Title:	**Window Overlapping**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-904	The FEE shall support overlapping windows.	

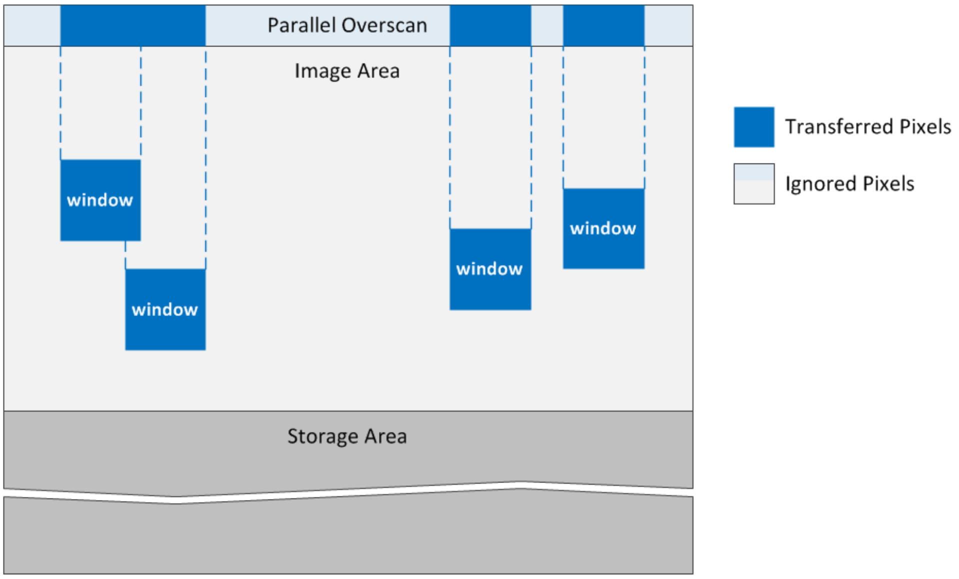
	Title:	**Parallel Overscan Transfer at N-FEE**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-566	The N-FEE shall be able to transmit complete lines from parallel overscan area in separate packets. The number of parallel overscan lines shall be configurable in a range from 0 to 31.	

Note: The overscan-lines will be used for the smearing correction. For this purpose lines on the top of the actual image will be shifted out the CCD.

	Title:	**Parallel Overscan Transfer at F-FEE**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-567	The F-FEE shall be able to transmit parts from parallel overscan area in separate packets. The number of parallel overscan lines shall be configurable in a range from 0 to 10. The F-FEE shall transmit only columns, which are a vertical projection of a window. The F-FEE shall derive the relevant columns from the window-list.	

Note: At the F-FEE at maximum 10 parallel overscan lines are accessible, because of the structure of the CCD. The first 5 lines will be dark lines and second 5 lines will be actual parallel overscan lines, if lines 0..4 are dumped during shift from the image into the storage area.

Figure: The following figure illustrates the transferred pixel from the parallel overscan in the F-FEE



	Title:	**Maximum Parallel Overscan Pixels (F-FEE)**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-934	The F-FEE shall be able to transfer 50% of the parallel overscan pixels for each CCD half.	

7 Configuration Interface

	Title:	**RMAP for Configuration and Status**
	Verif.:	Review-of-Design
FEE-DPU-IF-570	A subset of the remote-memory-access-protocol (RMAP), as defined in AD22, shall be used to configure the FEE and to gather status or housekeeping-information from the FEE.	

Note: The FEE may implement the full RMAP protocol, but only RMAP requests and replies described in this chapter shall be supported.

	Title:	**FEE Register Interface**
	Verif.:	Review-of-Design
FEE-DPU-IF-572	The FEE-configuration and the FEE-status shall be accessed by registers via RMAP. This means, for each function, parameter or status a dedicated address shall be specified. The register shall be plain. I.e. there will be no support for any kind of queues or buffers on a single address. The content of buffers shall be fully mapped into the RMAP address range.	

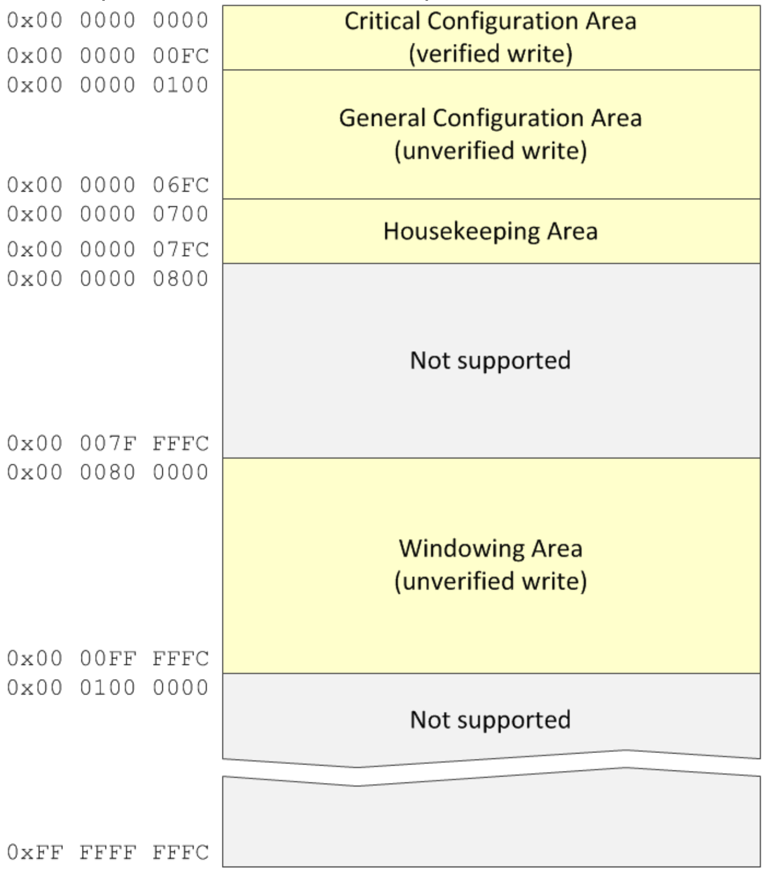
	Title:	**Read Access to Writable Bits**
	Verif.:	Review-of-Design
FEE-DPU-IF-953	Each writable bit in the register-interface shall be readable and shall reflect the write-contents.	

	Title:	**RMAP Memory Alignment**
	Justif.:	The DPUs are 32-bit systems.
	Verif.:	Review-of-Design
FEE-DPU-IF-573	The address of a DPU RMAP-request shall be 32-bit aligned. I.e. the address and the size of a RMAP request shall be a multiple of 4.	

	Title:	**RMAP Data Byte Encoding**
	Justif.:	The LEON-CPU is big-endian, so the complete PLATO payload shall be big-endian.
	Verif.:	Review-of-Design
FEE-DPU-IF-574	The encoding of 32-bit words shall be big-endian, so the most significant byte (MSB) shall be sent first and the least significant byte (LSB) shall be sent last.	

	Title:	**FEE Address Map**
	Verif.:	Review-of-Design
FEE-DPU-IF-575	The FEE memory-map shall be divided into seperate areas. Different restrictions for the RMAP-access shall be applicable for each memory-area. The following types or memory-areas shall be used: - critical configuration area (verify before write) - general configuration area - housekeeping area (read only) - windowing area	

Figure: Example of a FEE address map



Note: The FEE-teams are responsible for the detailed address map and the register assignment. The registers are described in the ICDs. A area-type can be used at multiple location of the memory.

	Title:	**Critical Configuration Areas**
	Verif.:	Review-of-Design
FEE-DPU-IF-578	The DPU shall use the verifiy-before-write option for RMAP write requests to a critical-configuration-area. All RMAP-request (read and write) to the critical-configuration-areas shall have a fixed data-length of 4 bytes.	

Note: It is recommended to use the critical-configuration-area for mode-settings, power-switching or settings with direct influence on the hardware. The critical configuration area can contain read-only registers.

	Title:	**General Configuration Areas**
	Verif.:	Review-of-Design
FEE-DPU-IF-579	The DPU shall disable the verifiy-before-write option for RMAP write-requests to a general-configuration-area. All RMAP-request (read and write) to the general-areas shall have a maximum data-length of 256 bytes.	

Note: It is recommended to use the general configuration area for non-critical configuration, like CCD timing settings.

	Title:	**Housekeeping Areas**

	Verif.:	Review-of-Design
FEE-DPU-IF-580	Housekeeping areas shall be read-only. Write requests to this area shall be ignored. All RMAP read-request to the HK-areas shall have a maximum data-length of 256 bytes.	

	Title:	**Windowing Areas**
	Verif.:	Review-of-Design
FEE-DPU-IF-581	The DPU shall disable the verifiy-before-write option for RMAP write-requests to a windowing-area. All RMAP-request (read and write) to the windowing-areas shall have a maximum data-length of 4096 bytes.	

	Title:	**RMAP Verified Write Request**
	Justif.:	AD22 chapter 5.3.1
	Verif.:	Test
FEE-DPU-IF-582	A RMAP-write-request to the critical configuration area (with verify-before-write option) shall use the following packet format.	

Figure: RMAP write request packet for the critical configuration area

0	logical address = 0x51
1	protocol id = 0x01
2	instruction = 0x7C
3	key = 0xD1
4	initiator address = 0x50
5	transaction id (MSB)
6	transaction id (LSB)
7	ext. address
8	address (MSB)
9	address
10	address
11	address (LSB)
12	data length (MSB) = 0x00
13	data length = 0x00
14	data length = 0x04
15	header CRC
16	data (MSB)
17	data
18	data
19	data (LSB)
20	data CRC

	Title:	**Verified Write– Instruction Field**
	Justif.:	AD22 chapter 5.1.4
	Verif.:	Test
FEE-DPU-IF-583	The DPU shall use RMAP instruction 0x7C for a write request to the critical configuration area.	

Note: According to AD22 the instruction has the following content:
- Bits 7:6 = b01, for RMAP request
- Bits 5:2 = b1111, for “write, incrementing address, verify before write, send reply”
- Bits 1:0 = b00, for length of reply address field is 0

	Title:	**RMAP Unverified Write Request**
	Justif.:	AD22 chapter 5.3.1
	Verif.:	Test
FEE-DPU-IF-585	A RMAP-write-request to a general-configuration-area or a windowing-area shall have the following packet format.	

Figure: RMAP write request packet without verify-before-write

0	logical address = 0x51
1	protocol id = 0x01
2	instruction = 0x6C
3	key = 0xD1
4	initiator address = 0x50
5	transaction id (MSB)
6	transaction id (LSB)
7	ext. address
8	address (MSB)
9	address
10	address
11	address (LSB)
12	data length (MSB)
13	data length
14	data length (LSB)
15	header CRC
16	data (MSB first)
N	data CRC

	Title:	**Unverified Write – Instruction Field**
	Justif.:	AD22 chapter 5.1.4
	Verif.:	Test
FEE-DPU-IF-586	The DPU shall use RMAP instruction 0x6C for a write request to a general-configuration-area or a windowing-area.	

Note: According to AD22 the instruction has the following content:
- Bits 7:6 = b01, for RMAP request
- Bits 5:2 = b1011, for "write, incrementing address, do not verify before write, send reply"
- Bits 1:0 = b00, for length of reply address field is 0

	Title:	**RMAP Read Request**
	Justif.:	AD22 chapter 5.4.1.1
	Verif.:	Test
FEE-DPU-IF-588	A RMAP-read-request shall have the following packet format.	

Figure: RMAP read request packet

0	logical address = 0x51
1	protocol id = 0x01
2	instruction = 0x4C
3	key = 0xD1
4	initiator address = 0x50
5	transaction id (MSB)
6	transaction id (LSB)
7	ext. address
8	address (MSB)
9	address
10	address
11	address (LSB)
12	data length (MSB)
13	data length
14	data length (LSB)
15	header CRC

	Title:	**RMAP Read – Instruction Field**
	Justif.:	AD22 chapter 5.1.4
	Verif.:	Test
FEE-DPU-IF-589	The DPU shall use RMAP instruction 0x4C for a read request.	

Note: According to AD22 the instruction has the following content:
- Bits 7:6 = b01, for RMAP request
- Bits 5:2 = b0011, for "read, incrementing address"
- Bits 1:0 = b00, for length of reply address field is 0

	Title:	**RMAP Request – Key Field**
	Justif.:	AD22 chapter 5.1.5
	Verif.:	Test
FEE-DPU-IF-591	The key-field in a RMAP request shall be 0xD1.	

	Title:	**RMAP Request – Initiator Address Field**

	Justif.:	AD22 chapter 5.1.7
	Verif.:	Test
FEE-DPU-IF-592	The initiator address field in a RMAP request shall be 0x50.	

	Title:	**RMAP Request – Transaction ID Field**
	Justif.:	AD22 chapter 5.1.8
	Verif.:	Test
FEE-DPU-IF-826	The DPU shall increment the transaction ID for each RMAP request.	

	Title:	**RMAP Request – Address Field**
	Justif.:	AD22 chapter 5.1.10
	Verif.:	Test
FEE-DPU-IF-827	The address field in a RMAP request shall contain the FEE register address. The extended address shall not be used and shall be zero.	

	Title:	**RMAP Request – Header and Data CRC Field**
	Justif.:	AD22 chapters 5.1.12 / 5.1.15 and ANNEX A
	Verif.:	Test
FEE-DPU-IF-828	The DPU shall calculate header and data CRC of RMAP requests as described in AD22.	

	Title:	**RMAP Reply**
	Justif.:	AD22 chapter 5.3.2.1
	Verif.:	Test
FEE-DPU-IF-832	The RMAP-reply packet to write-request shall have the following format:	

Figure: RMAP-reply packet to a write-request

0	logical address = 0x50
1	protocol id = 0x01
2	instruction
3	status
4	target address = 0x51
5	transaction id (MSB)
6	transaction id (LSB)
7	header CRC

	Title:	**RMAP Read Reply**
	Justif.:	AD22 chapter 5.4.2.2
	Verif.:	Test
FEE-DPU-IF-833	The RMAP-reply packet to read-request shall have the following format:	

Figure: RMAP-reply packet to a read-request

0	logical address = 0x50
1	protocol id = 0x01
2	instruction
3	status
4	target address = 0x51
5	transaction id (MSB)
6	transaction id (LSB)
7	reserved = 0
8	data length (MSB)
9	data length
10	data length (LSB)
11	header CRC
12	data (MSB first)
data-length + 12	data CRC

	Title:	**RMAP Reply – Logical Address**
	Justif.:	AD22 chapter 5.1.1
	Verif.:	Test
FEE-DPU-IF-834	The FEE shall put the initiator address of the RMAP request into the logical-address field of the RMAP reply packet.	

	Title:	**RMAP Reply – Instruction Field**
	Justif.:	AD22 chapter 5.1.4
	Verif.:	Test
FEE-DPU-IF-835	The FEE shall fill instruction field of the RMAP-reply with the following content: - Bits 7:6 shall be set to b00 to indicate a reply-packet. - Bits 5:2 shall contain the command from the request-packet. - Bits 1:0 shall contain the reply-address length from the request-packet.	

	Title:	**RMAP Reply – Status Field**
	Justif.:	AD22 chapter 5.1.17
	Verif.:	Test
FEE-DPU-IF-836	The FEE shall write 0 into the status-field of the RMAP-reply, if the command execution was successful.	

Note: The FEE shall either discard RMAP requests or reply with non-zero status according to AD22. The FEE shall support only the error-codes specified in this document.

	Title:	**RMAP Reply – Target Field**

	Justif.:	AD22 chapter 5.1.2
	Verif.:	Test
FEE-DPU-IF-837	The FEE shall write 0x51 into the target address field of the RMAP-reply.	

	Title:	**RMAP Reply – Transaction ID Field**
	Justif.:	AD22 chapter 5.1.8
	Verif.:	Test
FEE-DPU-IF-838	The FEE shall copy the transaction ID of the RMAP request into the transaction ID field of the RMAP-reply.	

	Title:	**RMAP Read Reply – Data Length Field**
	Justif.:	AD22 chapter 5.1.11
	Verif.:	Test
FEE-DPU-IF-839	The FEE shall copy the data-length of the RMAP request into the data-length field of the RMAP-reply.	

	Title:	**RMAP Reply – Header and Data CRC Field**
	Justif.:	AD22 chapters 5.1.12 / 5.1.15 and ANNEX A
	Verif.:	Test
FEE-DPU-IF-840	The FEE shall calculate the header and data CRC of RMAP replies as described in AD22.	

	Title:	**RMAP Reply Period**
	Justif.:	Needed for re-send mechanism.
	Verif.:	Test
FEE-DPU-IF-842	The FEE shall start sending the RMAP-reply within 10 milliseconds after the end of the request-packet.	

	Title:	**RMAP Write Across Memory Borders**
	Justif.:	A memory access across borders shall not be issued by the DPU and can be considered as failure.
	Verif.:	Test
FEE-DPU-IF-844	The FEE shall discard RMAP requests crossing a memory border.	

Note: The FEE-teams define the memory-map and the memory-borders.

	Title:	**RMAP Write to Unused Addresses**
	Verif.:	Test
FEE-DPU-IF-954	The FEE shall report RMAP write-requests to unused addresses as successful (status = 0).	

	Title:	**RMAP Read from Unused Addresses**
	Verif.:	Test
FEE-DPU-IF-845	The FEE shall report RMAP read-requests to unused addresses as successful (status = 0) and shall return a fixed pattern as data.	

	Title:	**Open RMAP requests**

	Justif.:	Because of the limitation to one request, the RMAP target does not need a request queue.
	Verif.:	Test
FEE-DPU-IF-936	If the DPU has sent a RMAP read request to a FEE, the DPU shall wait for the RMAP reply before sending a new RMAP request to the same device. After a time-out the DPU is allowed to send another request (see FEE-DPU-IF-863).	

	Title:	**RMAP-FDIR – Invalid RMAP Request Header**
	Justif.:	AD22 chapters 5.3.3.4.5 / 5.4.3.4.5
	Verif.:	Test
FEE-DPU-IF-846	The FEE shall discard RMAP requests, if the RMAP header is incomplete or the header CRC check fails.	

Note: The DPU will retry to send the RMAP request after a time-out.

	Title:	**RMAP-FDIR – EEP in Data Field**
	Verif.:	Test
FEE-DPU-IF-941	The FEE shall discard a RMAP request, if it was ended with an EEP.	

	Title:	**RMAP-FDIR – Invalid Data CRC in Request**
	Justif.:	AD22 chapters 5.3.3.6.5
	Verif.:	Test
FEE-DPU-IF-847	The FEE shall reply with status-code 4, if the data CRC check for a write request fails.	

Note: If the "verify before write" option is not used, the data will be written even if the request was rejected.

	Title:	**RMAP-FDIR – Invalid Key**
	Verif.:	Test
FEE-DPU-IF-848	The FEE shall discard a RMAP requests, if the key-field does not contain 0xD1.	

	Title:	**RMAP-FDIR – Invalid Target Address**
	Verif.:	Test
FEE-DPU-IF-849	The FEE shall discard a RMAP requests, if the logical address is not 0x51.	

	Title:	**RMAP-FDIR – Invalid Protocol ID**
	Justif.:	RMAP is the only Spacewire protocol for AEUs and MEU-PSUs.
	Verif.:	Test
FEE-DPU-IF-850	The FEE shall discard Spacewire packets with a protocol-ID other than 0x01.	

	Title:	**RMAP-FDIR – Invalid Command Code**
	Justif.:	AD22 chapters 5.3.3.5.4 / 5.4.3.5.4
	Verif.:	Test
FEE-DPU-IF-852	The FEE shall discard RMAP request if the request instruction is not supported by the FEE for the requested target address.	

Note: Only RMAP requests with instruction field 0x7C, 0x6C and 0x0C must be supported by the FEEs, depending on the memory area.

	Title:	**RMAP-FDIR – More or Less Data Than Expecpted**
	Verif.:	Test
FEE-DPU-IF-943	If the FEE shall discard RMAP write requests if more or less data are received than specified in the lenght-field.	

	Title:	**RMAP-FDIR – Unsupported Data Length**
	Verif.:	Test
FEE-DPU-IF-853	The FEE shall discard RMAP requests with unsupported data length.	

	Title:	**RMAP-FDIR – Invalid Length Alignment**
	Verif.:	Test
FEE-DPU-IF-855	The FEE shall discard RMAP requests if the value in the data-length field is not aligned to 32-bit.	

	Title:	**RMAP-FDIR – Early EOP**
	Verif.:	Test
FEE-DPU-IF-856	The DPU shall discard a RMAP reply, if it receives an incomplete header or less data than announced in the length-field. These failures shall be considered as early EOP.	

	Title:	**RMAP-FDIR – Too Much Data in Reply**
	Verif.:	Test
FEE-DPU-IF-949	The DPU shall discard a RMAP reply, if it more data than announced in the length-field.	

	Title:	**RMAP-FDIR – Wrong Data Length**
	Verif.:	Test
FEE-DPU-IF-966	The DPU shall discard RMAP read replies, if the reply contain more or less data than requested.	

	Title:	**RMAP-FDIR – Invalid Header CRC in Reply**
	Justif.:	AD22 chapters 5.3.3.11 / 5.4.3.11
	Verif.:	Test
FEE-DPU-IF-858	The DPU shall discard a RMAP reply, if the header CRC is not correct.	

	Title:	**RMAP-FDIR – Invalid Data CRC in Reply**
	Justif.:	AD22 chapter 5.4.3.12
	Verif.:	Test
FEE-DPU-IF-859	The DPU shall discard a RMAP reply, if the data CRC is not correct.	

Note: The requirement is applicable only for read requests.

	Title:	**RMAP-FDIR – Invalid Target Address**
	Justif.:	AD22 chapters 5.3.2.7 and 5.4.2.8

	Verif.:	Test
FEE-DPU-IF-860	The DPU shall discard a RMAP reply, if the target address is not equal to 0x51.	

	Title:	**RMAP-FDIR – Invalid Status**
	Justif.:	RMAP request failed and shall be repeated.
	Verif.:	Test
FEE-DPU-IF-861	The DPU shall discard the RMAP reply, if the status field is non-zero.	

	Title:	**RMAP-FDIR – Invalid Transaction ID**
	Justif.:	AD22 chapters 5.1.8, 5.3.2.8 and 5.4.2.9
	Verif.:	Test
FEE-DPU-IF-862	The DPU shall discard the RMAP reply, if the transaction-ID in the reply is not equal to the transaction-ID of last RMAP request.	

	Title:	**RMAP-FDIR – Request Repeats**
	Justif.:	Recovery action for temporary failures.
	Verif.:	Test
FEE-DPU-IF-863	The DPU shall repeat the last RMAP-request after a time-out of the reply or if the status of the reply was non-zero. The time-out between the retries shall be configurable in a range of 0-10 seconds with at least 100ms steps. The maximum number of retries shall be configurable in a range of 0..31.	

	Title:	**RMAP-FDIR – DPU Error Counter**
	Justif.:	Failure monitoring.
	Verif.:	Test
FEE-DPU-IF-864	The DPU shall contain an error-counter for each FEE, which shall be incremented on any kind of RMAP error (time-out, reply-status not 0, invalid reply). The error-counter shall increment only once per packet, even if the packet contains multiple errors.	

	Title:	**RMAP-FDIR – DPU Error Report**
	Justif.:	RMAP failures are expected to be rare. Reporting of the last error is sufficient. The error counter shows if more failures occured.
	Verif.:	Test
FEE-DPU-IF-865	The DPU shall report the following information about the last RMAP error in the housekeeping-data: - Time-out error and number of retries for this request. - Reply status field, if the status is non-zero. - CRC check error in header or data. - Invalid header fields, including the information which field was corrupted - The reception of EEP, early EOP or more data than expected.	

8 Data Interface

	Title:	**Data Packet Format**
	Verif.:	Test
FEE-DPU-IF-594	For the transfer of image- and housekeeping-data from FEE to DPU a proprietary packet-format shall be used. The FEE data-packet consists of a 10 byte header and a data-field with variable length.	

Figure: Data and HK packet-structure, including Spacewire-address and protocol-ID

0	logical address = 0x50
1	protocol id = 0xF0
2	length (MSB)
3	length (LSB)
4	type (MSB)
5	type (LSB)
6	frame counter (MSB)
7	frame counter (LSB)
8	sequence counter (MSB)
9	sequence counter (LSB)
10	data (big endian)
length+9	

	Title:	**Data Packet Header**
	Verif.:	Test
FEE-DPU-IF-871	The first two bytes of the data-packet are the logical address and the protocol-ID regarding AD20. Bytes 2 to 8 contain the header of the data packet with the following content: - 16-bit data-length, given in number of bytes - 16-bit type - 16-bit frame-counter - 16-bit sequence-counter	

	Title:	**Data Packet Byte Encoding**
	Verif.:	Review-of-Design
FEE-DPU-IF-872	The encoding of 16-bit or 32-bit words shall be big-endian, so the most significant byte (MSB) is in lower address and the least significant byte (LSB) in the higher address. The endianness is applicable for header and data field.	

	Title:	**N-FEE Packet Length**
	Verif.:	Test, Review-of-Design

FEE-DPU-IF-873	The data packets length of the N-FEE shall be configurable in range between 1024 and 32768 bytes (header + data field) in all modes. This length shall be fixed for the read-out cycle. If there is not enough data at the end of the read-out to fill a complete packet, the last packet might be shorter.

Note: For instance, the length of 18370 bytes will be choosen in full-image mode to hold 4 complete lines in a packet (10 byte header + 2 bytes * (4560 + 30) pixels * 4 lines).

Note: Because a pixel has size of 16 bit, the length must be always aligned to 16-bit.

	Title:	**F-FEE Packet Length in Full-Image Mode**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-967	The data packets length of the F-FEE shall be configurable in range between 1024 and 32768 bytes (header + data field) in full-image, full-image-pattern and partial read-out mode. This length shall be fixed for the read-out cycle. If there is not enough data at the end of the read-out to fill a complete packet, the last packet might be shorter.	

	Title:	**F-FEE Packet Length in Windowing-Modes**
	Justif.:	The F-FEE does not have an external memory and the block-RAMs inside the FPGA are limited.
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-874	In windowing-mode and windowing-pattern-mode the F-FEE data-packets shall have a fixed size. Only the last packet of the image can be shorter. The minimum length of the data packet shall be 128 bytes (header + data field).	

	Title:	**Spacewire Protocol ID for FEE-Data**
	Verif.:	Test
FEE-DPU-IF-876	In PLATO the protocol-ID 0xF0 shall be used for FEE data packets.	

Note: Regarding to AD21, chapter 5.2.5, the protocol IDs 0xF0 to 0xFE can be defined by the project.

	Title:	**Data Packet Field: Length**
	Verif.:	Test
FEE-DPU-IF-877	Bytes 2 and 3 of the data-packet-header contain the data-length in bytes.	

	Title:	**Data Packet Field: Type**
	Verif.:	Test
FEE-DPU-IF-878	Bytes 4 and 5 of the data-packet-header contains additional information about the packet-content. The type-field is defined in the following way: - bits 15:11 = reserved for future usage - bit 10:8 = mode: 0 = full-image mode, 1= full-image pattern mode, 2 = windowing mode, 3 = windowing pattern mode, 4 = partial read-out mode - bit 7 = last packet: 1 = last packet of the this type in the current read-out-cycle - bit 6 = CCD side: 0 = left side (side E), 1 = right side (side F) - bits 5:4 = CCD number: tbd. - bits 3:2 = frame number after sync - bits 1:0 = packet type: 0 = data packet, 1 = overscan data, 2 = housekeeping packet	

Note: Because the data for left and right CCD-side is send in different packets, there will be a last packet for left CCD-side and a last packet for the right CCD-side. Also the last packet with HK or overscan data must contain the last-packet-flag.
Note: As the CCD sequence is not fixed for the N-Cameras, the frame number was introduced. The frame number is set to 0 when the 25 sec sync-impulse was received and it incrementents at each 6.25 sec sync-impulse. For the F-FEE the number can be set to zero.

	Title:	**Data Packet Field: Frame Counter**
	Verif.:	Test
FEE-DPU-IF-880	The frame-counter shall be incremented after every full CCD read-out cycle, i.e. in the N-DPU every 25 seconds, in the F-DPU every 2.5 seconds.	

	Title:	**Data Packet Field: N-FEE Frame Number**
	Verif.:	Test
FEE-DPU-IF-968	In all modes, the N-FEE shall update the frame number register (readable by RMAP) as follows: The N-FEE shall set the frame number to 0, when the 25 sec sync-impulse was received, and it shall increment the frame-number at each 6.25 sec sync-impulse.	

	Title:	**Data Packet Field: Sequence Counter**
	Verif.:	Test
FEE-DPU-IF-881	The FEE shall have a sequence-counter for each CCD. The sequence-counter shall be set to zero at beginning of every CCD-read-out. The sequence counter shall be applicable for HK and image data packets.	

Note: Because HK packets are send before the image data, the first HK packet contains the sequence-counter value of 0.

	Title:	**Sequence Counter Consistency-Check**
	Verif.:	Test
FEE-DPU-IF-882	Before window-assembly the DPU shall check the sequence-counter of the received packets to confirm the expected order of the packets in the memory.	

	Title:	**Data Packet Field: Data**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-883	Depending on the type-field, the data-field contains either image-data or housekeeping-data.	

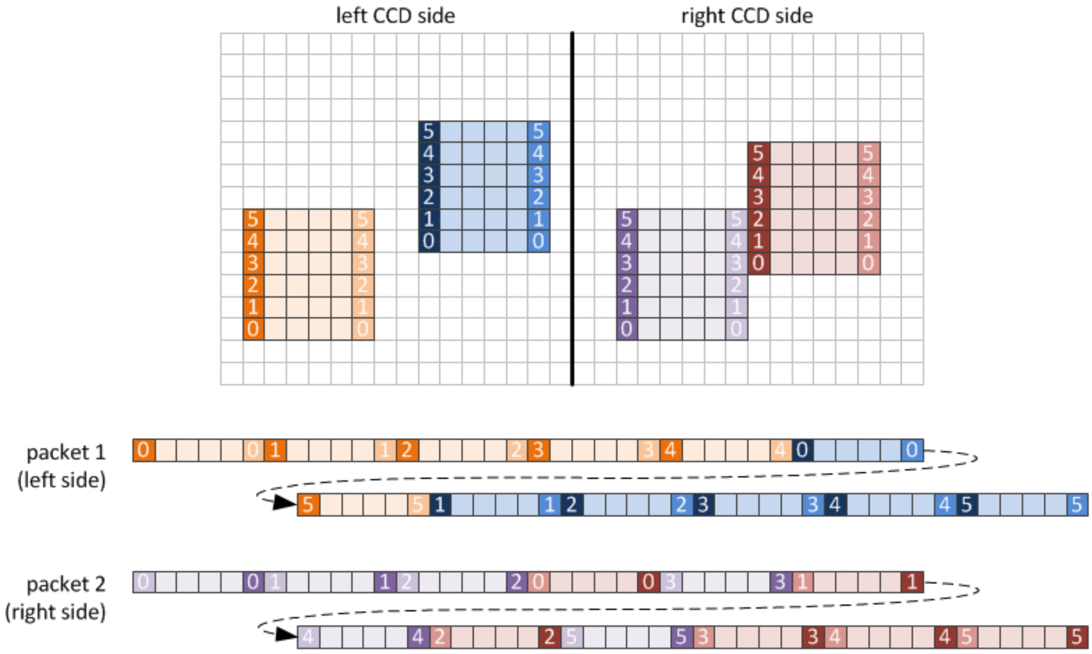
	Title:	**Image Data Format**
	Verif.:	Test
FEE-DPU-IF-884	The image data is transferred as 16-bit integer values, each value representing one pixel.	

	Title:	**CCD-Side Data-Separation**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-885	The data of the right- and left CCD-side shall be sent in separate packets.	

	Title:	**Data Read-Out Order**

	Verif.:	Test
FEE-DPU-IF-901	The data should be transferred in the order of the CCD read-out.	

Figure: Example for data-order in windowing-mode



	Title:	**Data Transfer Consistency**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-886	The data transfer from the FEE to the DPU shall be deterministic for a specific set of windows. So, the order of the packets shall be the identical for every transfer. Especially the order of the left and right CCD-side packets shall be consistent over consecutive data-transfers.	

Motivation: For a specific set of windows the DPU will prepare a list of copy-operation. This copy-list will help to quickly assemble the windows for further processing. If the packets would arrive in random order, the copy-operations must be calculated in real-time and the assembly-operation would take much longer.

	Title:	**Housekeeping Data Format**
	Verif.:	Review-of-Design
FEE-DPU-IF-888	The format, i.e. the position of each HK-value, of the N-FEE and F-FEE housekeeping data shall be fixed.	

Note: The FEE may generate several HK packets. But the structure must correspond always with the sequence number.

	Title:	**HK-Packet Generation Period**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-889	At every CCD-readout a HK packet shall be generated. This means the HK packet will be send every 6.25 seconds (N-FEE) or 2.5 seconds (F-FEE).	

	Title:	**HK-Packet Position**
	Verif.:	Test
FEE-DPU-IF-890	The HK packet shall be send before the image data.	

	Title:	**Data FDIR – Sequence Check Failed**
	Verif.:	Test
FEE-DPU-IF-891	If the sequence counter has not the expected value, the DPU shall dump the corresponding packets.	

	Title:	**Data FDIR – EEP**
	Verif.:	Test
FEE-DPU-IF-892	If an EEP occurs, the F-DPU shall dump the corresponding packet.	

	Title:	**Data FDIR – DPU Error Counter**
	Verif.:	Test
FEE-DPU-IF-893	The DPU shall contain an error-counter, which shall be incremented on any kind of data error.	

	Title:	**Data FDIR – DPU Error Report**
	Verif.:	Test
FEE-DPU-IF-894	The DPU shall report every data error in the housekeeping-data with an unambiguous code.	

9 Pattern Generation

	Title:	**Pattern Generation**
	Verif.:	Test, Review-of-Design
FEE-DPU-IF-896	If the FEE is full-image-pattern-mode or in windowing-pattern-mode, the FEE shall send artificial data pattern instead of CCD-Data. The pattern shall be the same for windowing and full-image mode.	

	Title:	**Pattern Structure**
	Verif.:	Test
FEE-DPU-IF-897	The data pattern shall have the following structure: - Bits [15:13] = time-code % 8 - Bits [12:11] = CCD number - Bit [10] = CCD side: 0 = left side, 1 = right side - Bit [9:5] = X-coordinate % 32 - Bit [4:0] = Y-coordinate % 32 The details of pattern are defined in RD02.	

Note: For a dedicated coordinate each pixel has the same content in windowing-pattern-mode and in full-image-pattern-mode.

Figure: structure of the data pattern

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timecode [2:0]			CCD		side	Row [4:0]					Column [4:0]				

10 APPENDIX