Title

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

Subtitle **Issue for Software SRR / iPDR**

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

Issue **Draft B**

Date 23.10.2017

---	Name	Date	Signature
Prepared:	Karsten Westerdorff, DLR	20.10.2017	
Checked:	Class Ziemke, DLR		
---	Bernd Ulmer, DLR/IBU		
---	Alexander Koncz, DLR		
---	Philippe Plasson, LESIA		
---	Tom Kennedy, MSSL		
---	Matthias Wieser, OHB		
Approved:	Peter Becker, DLR		
Authorized:	Gisbert Peter, DLR		

CHANGE HISTORY

Issue	Change	Approved	Date
1	**Initial release**	K.Westerdorff	05.04.2017
1.1	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	23.10.2017
.			
.			
.			
.			

EXPORTED MODULES FROM IBM DOORS

Module	Module ID
/PLATO/Payload/Software/FEE-DPU IRD	**0000036e**

Exported Version	Current Version, Last change: 23.10.2017 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 - 36

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	N-FEE URD (User Requirement Document)	PLATO-OHB-PL-RS-0005, Issue 3, Draft 1 (03/2017)
AD02	F-FEE URD (User Requirement Document)	PLATO-OHB-PL-RS-0005, Issue 3, Draft 1 (03/2017)
AD03	MEU URD (User Requirement Document)	PLATO-DLR-PL-RS-003, Issue 4, Draft 5 (03/2017)
AD04	FEU URD (User Requirement Document)	PLATO-DLR-PL-RS-004, Issue 4, Draft (03/2017)
AD05	PLATO Software System Specification	PLATO-DLR-PL-RS-006, Issue 1.1, Oct2015

2.2 Applicable ECSS Standards

	Title	Reference
AD20	SpaceWire - Links, nodes, routers and networks	ECSS-E-ST-50-12C
AD21	SpaceWire protocol identification	ECSS-E-ST-50-51C
AD22	SpaceWire - Remote memory access protocol	ECSS-E-ST-50-52C

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 with its FEE box attached to the rest of the camera by a temporary structure which shall be removed during the integration of the camera on the optical bench. 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. 4 N-AEU boxes provides voltages for the normal FEEs/cameras, one N-AEU for one batch of 8 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.

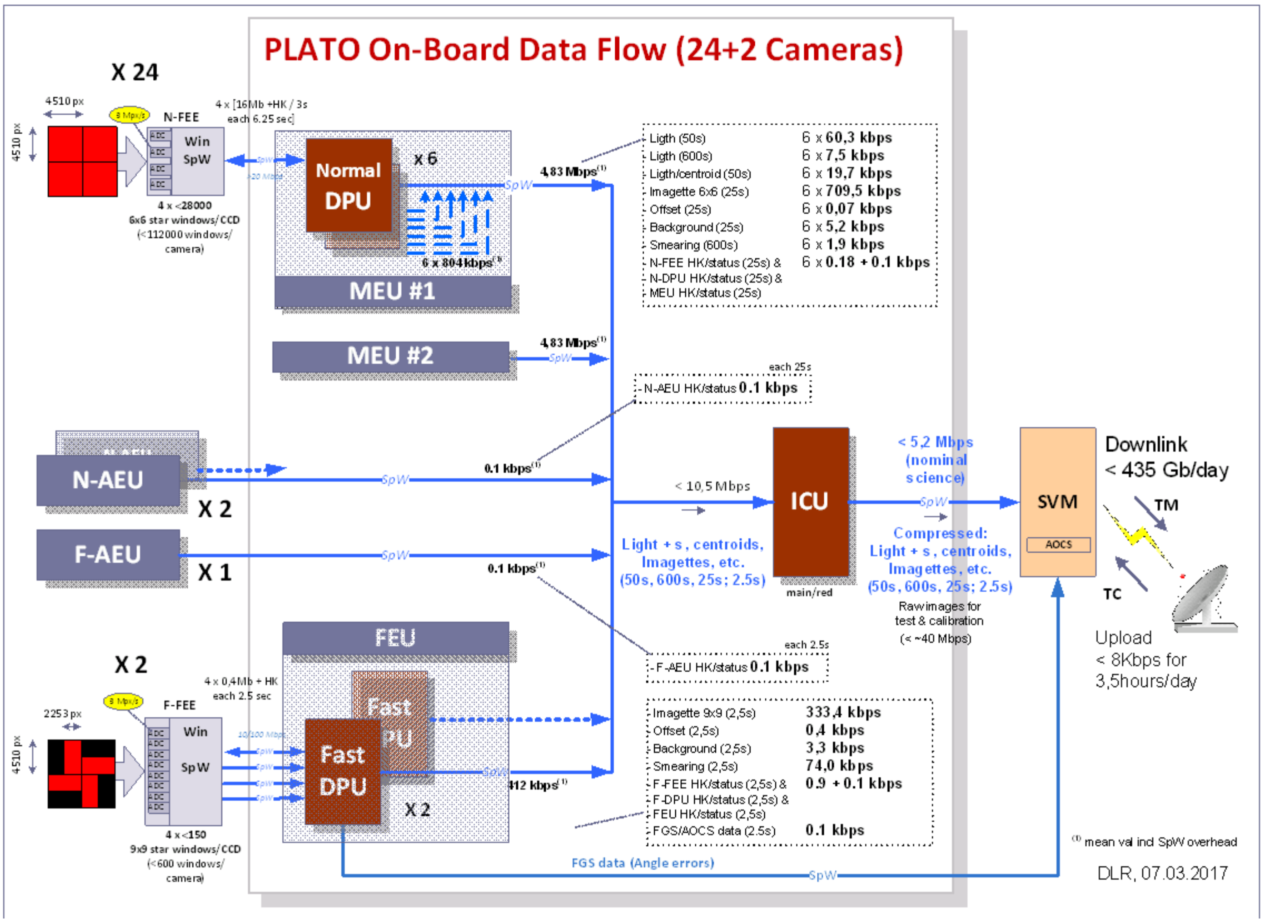
There are 2 fast DPUs gathered in one electronic box named FEU (Fast Electronic Unit). Each F-DPU is responsible for processing the data of one fast camera. The processing cadence for F-DPUs is 2.5 sec.

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

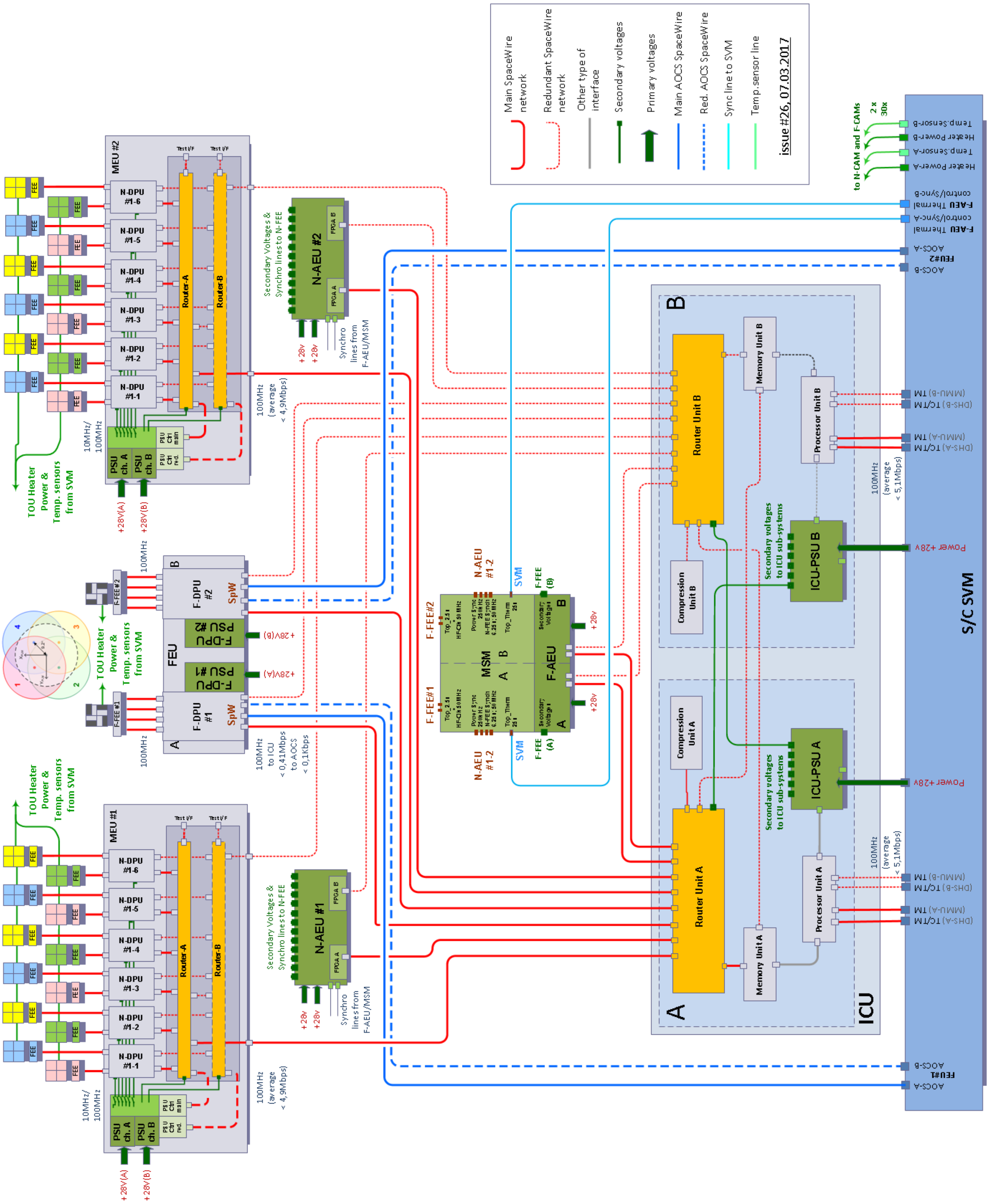


Due to fault tolerance reasons and in order to optimize the resources (mass, volume, harness), the physical implementation of the architecture described above foresees to split the 12 N-DPUs in 2 groups of 6 N-DPUs. Each group of 6 N-DPUs is gathered in a box called Main Electronic Unit (MEU).

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.

	Title: **Spacewire Interface**
FEE-DPU-IF-529	The electrical and mechanical parameters of the Spacewire interface shall be compliant to standard AD20.

	Title: **Grounding**
FEE-DPU-IF-530	TBD

	Title: **Spacewire Speed - Uplink**
FEE-DPU-IF-531	All Spacewire links between FEE and DPU shall run with 40 Mbps ± 5% in uplink (from DPU to FEE).

	Title: **Spacewire Speed - Downlink**
FEE-DPU-IF-532	All Spacewire links between FEE and DPU shall run with 100 Mbps ± 5% in downlink (from FEE to DPU).

	Title: **Spacewire Timing Margin**
FEE-DPU-IF-533	SpaceWire link elements shall be compliant to following budget: - Jitter + Skew of Encoder and associated drivers including unit connector: max. 27% of 1/SpW data-signal-rate - Jitter + Skew of Decoder and associated receivers including unit connector: max. 42% of 1/SpW data-signal-rate - Jitter + Skew of cable between units including: max. 21% of 1/SpW data-signal-rate

Note: A margin 10% of 1/SpW data-signal-rate is left for safety reason.

	Title: **Spacewire Link-Start**
FEE-DPU-IF-914	The DPU shall be the master on the Spacewire-interface. After power-up the Spacewire-interface of the FEE shall be in Autostart-Mode. The DPU shall be responsible for starting the Spacewire-link.

	Title: **Spacewire Error Register**
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**
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**
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**
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**
FEE-DPU-IF-539	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.

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

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

5 FEE-Modes

	Title: **FEE Modes**
FEE-DPU-IF-543	The FEE shall support the following modes: - Stand-By Mode - Calibration-Mode - Calibration-Pattern-Mode - Windowing-Mode - Windowing-Pattern-Mode - Partial-Readout-Mode

	Title: **Default Mode**
FEE-DPU-IF-544	After power-on the FEE shall be in stand-by-mode.

	Title: **Stand-By-Mode**
FEE-DPU-IF-545	The purpose of the stand-by-mode is to configure the FEE. Most of the configuration-settings may be locked outside the stand-by-mode. All other modes shall be entered from the stand-by-mode.

	Title: **Calibration-Mode**
FEE-DPU-IF-546	In the calibration-mode a half-CCD (N-FEE) or four half-CCDs (F-FEE) will be transmitted to the DPU. Because the bandwidth to the DPU is limited, a complete CCD-image cannot be transmitted at once. Therefore the transmission is stretched over two cycles.

	Title: **Calibration-Pattern-Mode**
FEE-DPU-IF-925	In the calibration-pattern-mode is almost identical to the calibration-mode, but synthetic data pattern shall be transferred instead of actual CCD data.

	Title: **Windowing-Mode**
FEE-DPU-IF-547	In the windowing-mode the full CCD will be read out, but only regions of interests will be transmitted to the DPU.

	Title: **Windowing-Pattern-Mode**
FEE-DPU-IF-926	In the windowing-pattern-mode is almost identical to the windowing-mode, but synthetic data pattern shall be transferred instead of actual CCD data.

	Title: **Partial-Readout-Mode**
FEE-DPU-IF-548	In the partial readout mode most of the lines shall be dumped during readout to avoid over-saturation in ambient conditions.

	Title: **Mode Change**
FEE-DPU-IF-923	The mode shall be changed always on the external sync-signal.

	Title: **Emergency Stand-By**
FEE-DPU-IF-924	The FEE shall return to stand-by-mode immediately if a special command "emergency stand-by" is sent by the DPU.

	Title: **CCD Read Out Order (N-FEE)**
FEE-DPU-IF-937	The sequence of CCD read out in the N-FEE shall be configurable. For test purpose it shall be possible always the same CCD at every external sync impulse.

6 Windowing

	Title:	**Coordinate System and Orientation**
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

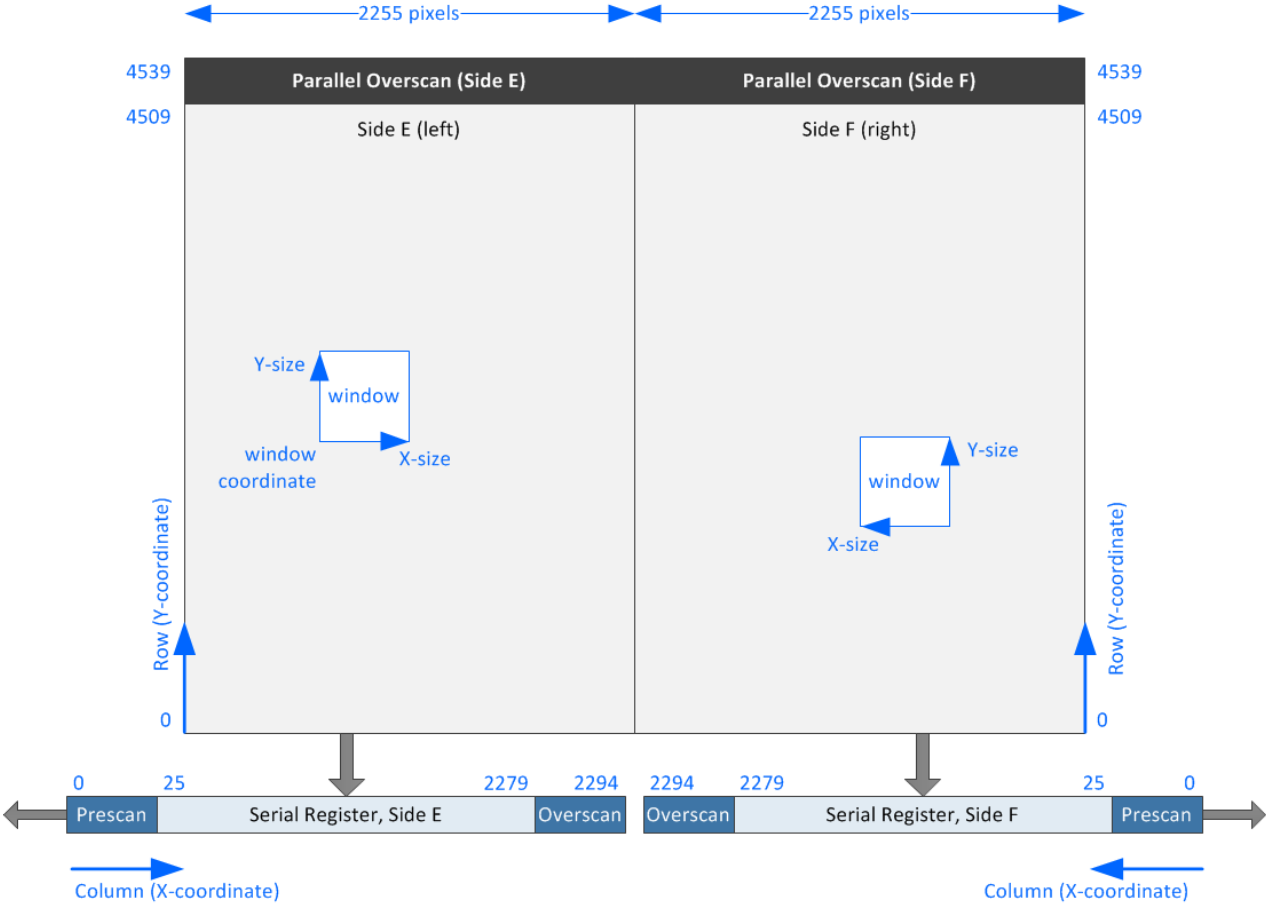
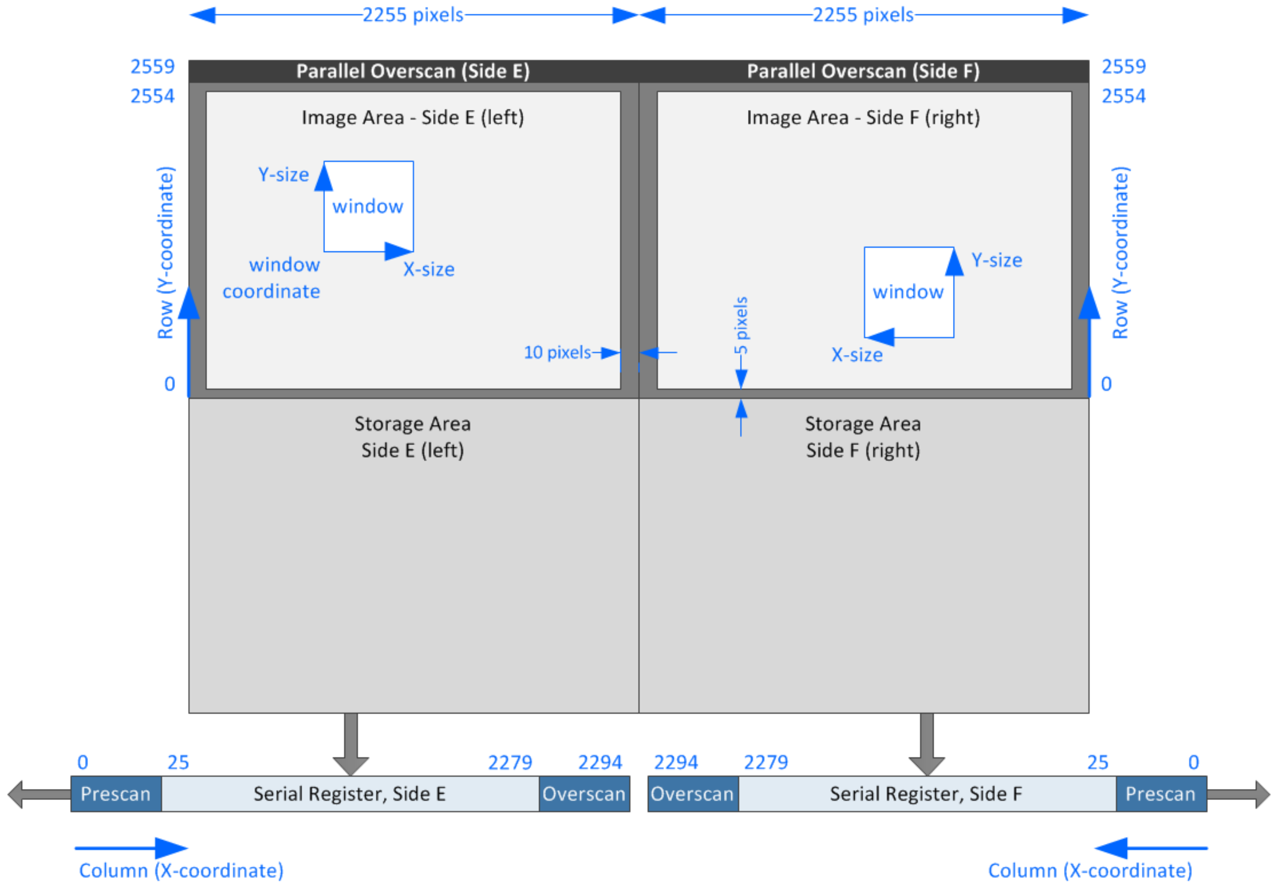


Figure: Coordinate-system of the F-camera and the orientation of the windows



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

Figure: Structure of the window parameters

15	13	12	0
01b	Y coordinate (row)		
10b	side	X coordinate (column)	
11b	active window position		

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.

Note: The active window position is optional parameter needed for a special implementation in the N-FEE. With this approach the position of the active window is preprocessed to save additional memory and logic in the N-FEE FPGA. It is recommend to generate a bit mask for the next line, similar to the F-FEE, without using the active window position.

	Title:	**Window Lists**

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 Size (N-FEE)**
FEE-DPU-IF-905	The N-FEE shall store up to 115,000 window-coordinates, summarized over the four window-lists.

	Title: **Window List Size (F-FEE)**
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)**
FEE-DPU-IF-921	The F-FEE shall be able to process 512 windows per CCD.

	Title: **Maximum Windows per Row (N-FEE)**
FEE-DPU-IF-919	The N-FEE shall be able to handle up to 766 active windows per line. This number results from a worst estimation, where 6 pixel wide windows cover alle pixels in the line without overlapping = (2255 pixels per line + 40 pre-/overscan) / 6 * 2

	Title: **Maximum Windows per Column (N-FEE)**
FEE-DPU-IF-920	The F-FEE shall be able to handle up to 128 windows per column. This number results from a technical worst-case estimation of having at maximum a quarter of the windows in one line.

	Title: **Window-List Pointer Registers**
FEE-DPU-IF-554	For each window-list there shall be two registers, which contain 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)**
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 and active-window-position (optional) follow. At a new row the next Y-coordinate is stated in the list.

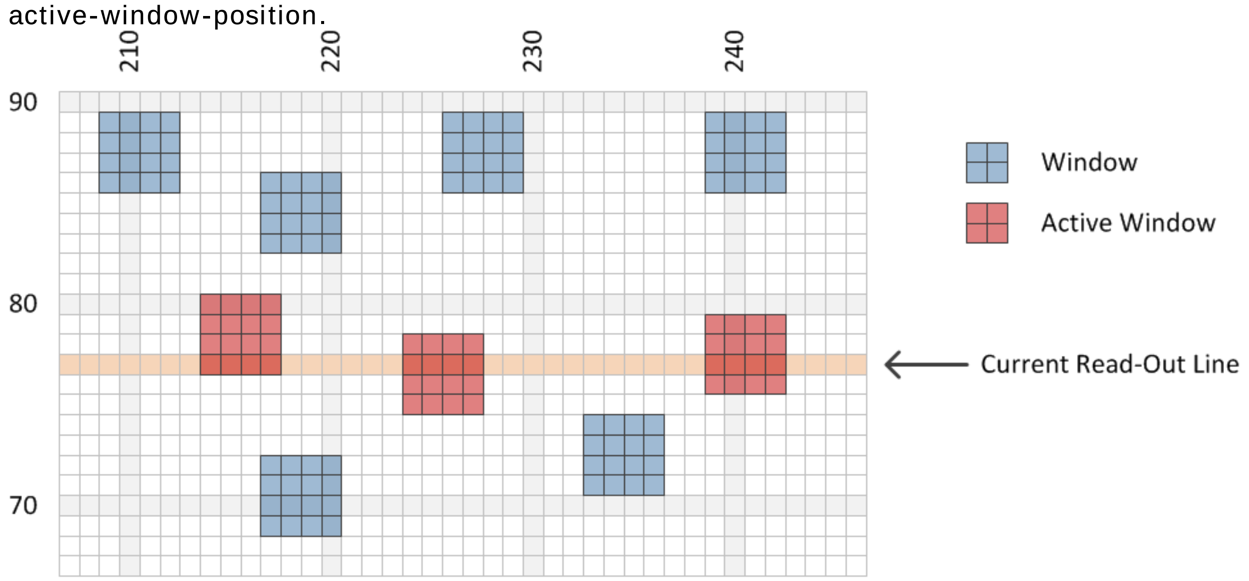
	Title: **Window-List Sorting (N-FEE)**
FEE-DPU-IF-558	The window-list for the N-FEE shall be sorted by the Y-coordinate (row) first and by the X-coordinate (column) second.

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

	Title: **Window-List Sorting (F-FEE)**
FEE-DPU-IF-561	The window-list for the F-FEE shall be sorted by the X-coordinate (column) first and by the Y-coordinate (row) second.

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. So it can prepare a bit-mask for the next line. 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 (Option A)	N-FEE (Option B)	F-FEE
4045h (Y: 4000h + 69d)	4045h (Y: 4000h + 69d)	80D1h (X: 8000h + 209d)
80D9h (X: 8000h + 217d)	80D9h (X: 8000h + 217d)	4056h (Y: 4000h + 86d)
4047h (Y: 4000h + 71d)	C000h (X: E000h + 0d)	80D6h (X: 8000h + 214d)
80E9h (X: 8000h + 233d)	4047h (Y: 4000h + 71d)	404Dh (Y: 4000h + 77d)
404Bh (Y: 4000h + 75d)	80E9h (X: 8000h + 233d)	80D9h (X: 8000h + 217d)
80E0h (X: 8000h + 224d)	C001h (X: E000h + 1d)	4045h (Y: 4000h + 69d)
404Ch (Y: 4000h + 76d)	404Bh (Y: 4000h + 75d)	4053h (Y: 4000h + 83d)
80EFh (X: 8000h + 239d)	80E0h (X: 8000h + 224d)	80E0h (X: 8000h + 224d)
404Dh (Y: 4000h + 77d)	C000h (X: E000h + 0d)	404Bh (Y: 4000h + 75d)
80D6h (X: 8000h + 214d)	404Ch (Y: 4000h + 76d)	80E2h (X: 8000h + 226d)
4053h (Y: 4000h + 83d)	80EFh (X: 8000h + 239d)	4056h (Y: 4000h + 86d)
80D9h (X: 8000h + 217d)	C001h (X: E000h + 1d)	80E9h (X: 8000h + 233d)
4056h (Y: 4000h + 86d)	404Dh (Y: 4000h + 77d)	4047h (Y: 4000h + 71d)
80D1h (X: 8000h + 209d)	80D6h (X: 8000h + 214d)	80EFh (X: 8000h + 239d)
80E2h (X: 8000h + 226d)	C000h (X: E000h + 0d)	404Ch (Y: 4000h + 76d)
80EFh (X: 8000h + 239d)	⋮	4056h (Y: 4000h + 86d)

FEE-DPU-IF-562	It shall be possible to change the window-list 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**
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.	

Note: The default imagette size for normal and fast cameras is 6x6 pixels. For the FGS a windows size of 9x9 may 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**
FEE-DPU-IF-565	In windowing-mode it shall be possible the change the window-size outside read-out.	

	Title:	**Window Overlapping**
FEE-DPU-IF-904	The FEE shall support overlapping windows.	

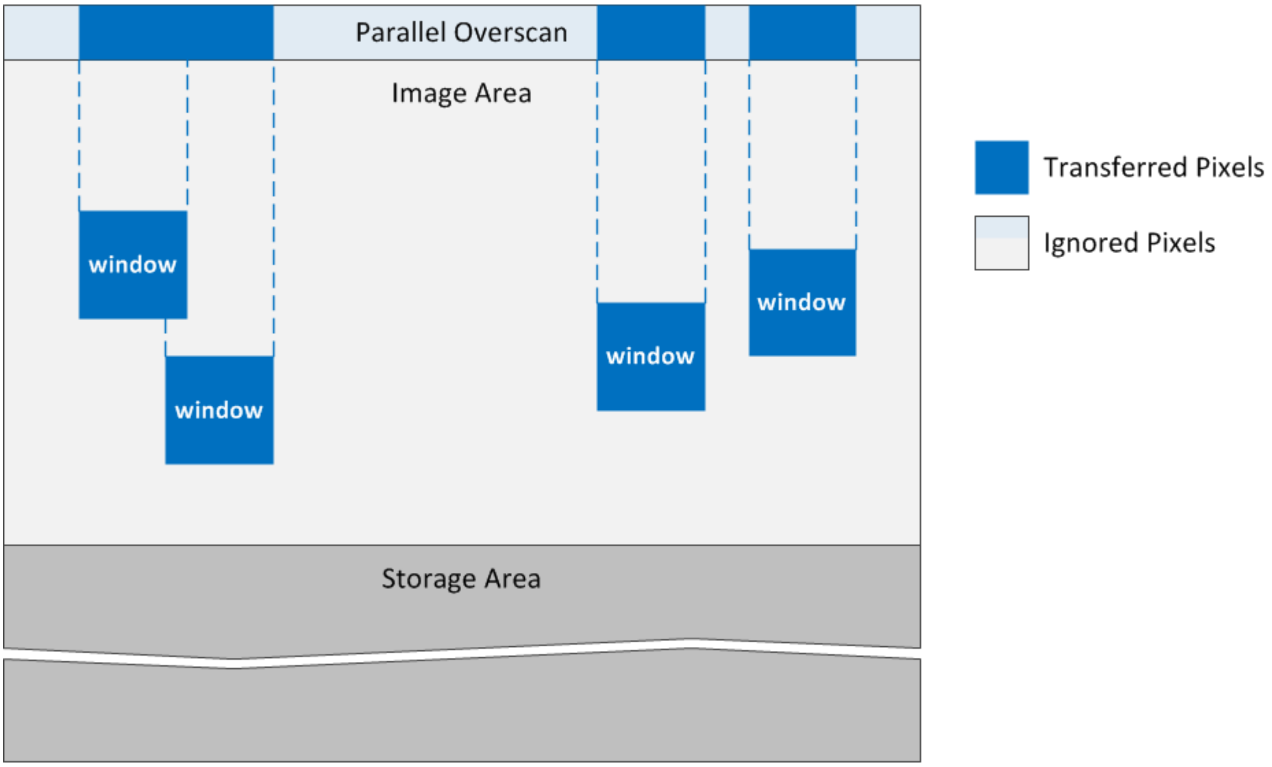
	Title:	**Parallel Overscan Transfer at N-FEE**
FEE-DPU-IF-566	The N-FEE shall be able to transmit complete lines from parallel overscan area in separate packets. The number of parallel overscan lines shall be configurable in a range from 0 to 31.	

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**
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)**
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**
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 must be supported.

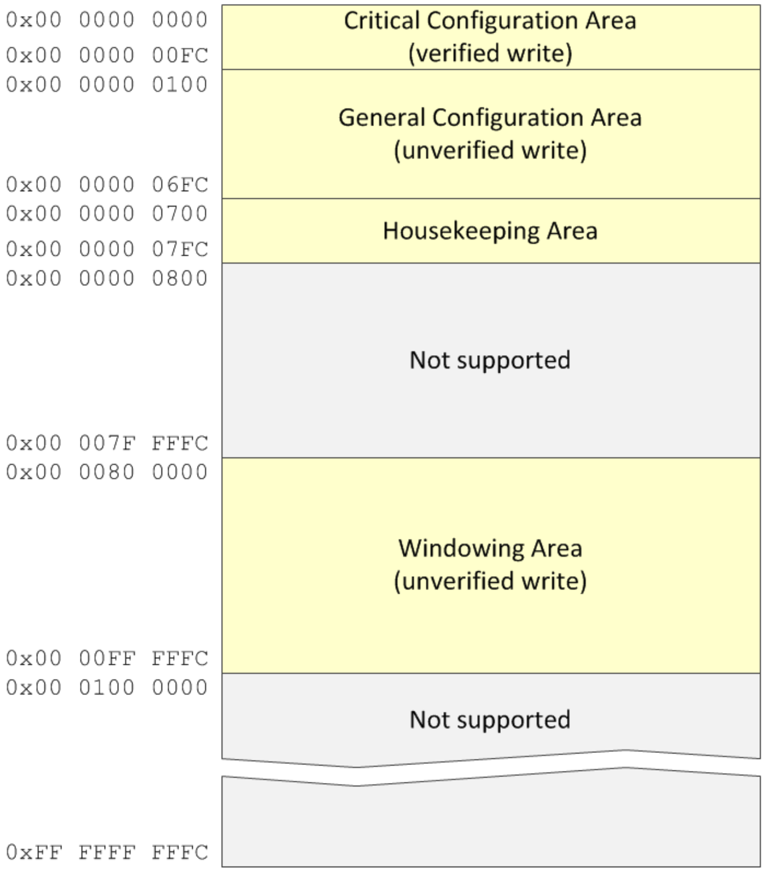
	Title:	**FEE Register Interface**
FEE-DPU-IF-572	The FEE-configuration and the FEE-status shall be accessed by registers via RMAP. This means, for each function, parameter or status a dedicated address shall be specified. The register shall be plain. I.e. there will be no support for any kind of queues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

	Title:	**RMAP Memory Alignment**
FEE-DPU-IF-573	The address for a RMAP-access shall be 32-bit aligned. A RMAP access shall have a size of multiple 32-bit words.	

	Title:	**RMAP Data Byte Encoding**
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**
FEE-DPU-IF-575	The RMAP memory shall be divided into areas for critical configuration, general configuration, housekeeping and windowing.	

Figure: Address map of the FEE



	Title:	**Critical Configuration Area**
FEE-DPU-IF-578	The address-range 0x0000_0000 to 0x0000_00FF shall be reserved for critical configuration settings, like FEE-mode, subsystem power, etc. A write access to this address-range shall be done by a RMAP verified-write-request. Registers in the critical configuration-area shall be readable by a RMAP read-request.	

	Title:	**General Configuration Area**
FEE-DPU-IF-579	The address-range 0x0000_0100 to 0x0000_06FF shall be reserved for non-critical configuration settings, like CCD timing settings. A write access to this address-range shall be done by a RMAP unverified-write-request. Registers in the general configuration-area shall be readable by a RMAP read-request.	

	Title:	**Housekeeping Area**
FEE-DPU-IF-580	The address-range 0x0000_0700 to 0x0000_07FF shall be reserved for housekeeping and status information. Registers in the general configuration-area shall be readable by a RMAP read-request.	

	Title:	**Windowing Area**
FEE-DPU-IF-581	The address-range 0x0080_0000 to 0x00FF_FFFF shall be reserved for the window-lists. A write access to this address-range shall be done by a RMAP unverified-write-request. Registers in the windowing-area shall be readable by a RMAP read-request.	

	Title:	**RMAP Verified Write Request**
FEE-DPU-IF-582	A RMAP-verified-write-request with a fixed data-length of 4 bytes shall be used by the DPU to set register-values in the critical configuration-area of the FEE.	

Figure: Request-packet for verified write access

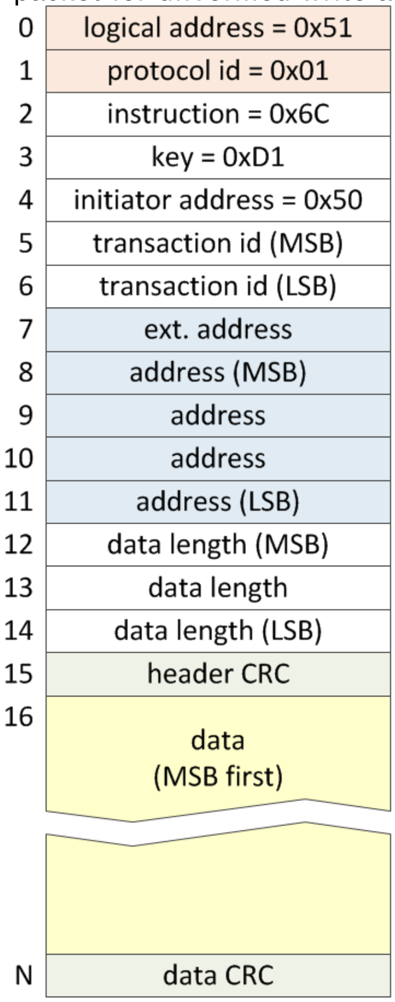
0	logical address = 0x51
1	protocol id = 0x01
2	instruction = 0x7C
3	key = 0xD1
4	initiator address = 0x50
5	transaction id (MSB)
6	transaction id (LSB)
7	ext. address
8	address (MSB)
9	address
10	address
11	address (LSB)
12	data length (MSB) = 0x00
13	data length = 0x00
14	data length = 0x04
15	header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

	Title:	**Verified Write– Instruction Field**
FEE-DPU-IF-583	For a write request to the critical configuration area, the RMAP instruction shall be 0x7C. Regarding AD22 this means: - 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:	**Verified Write–Length Field**
FEE-DPU-IF-584	The data length of a verified write request shall be fixed to 4 bytes.	

	Title:	**RMAP Unverified Write Request**
FEE-DPU-IF-585	A RMAP-verified-write-request shall be used by the DPU to set register-values in the general-configuration-area or in the windowing-area of the FEE.	

Figure: Request-packet for unverified write access



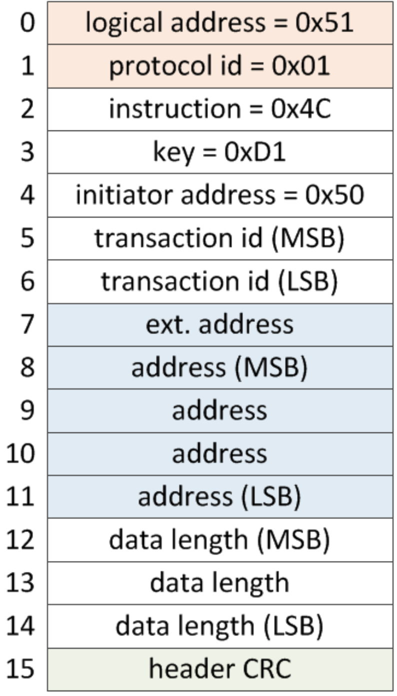
0	logical address = 0x51
1	protocol id = 0x01
2	instruction = 0x6C
3	key = 0xD1
4	initiator address = 0x50
5	transaction id (MSB)
6	transaction id (LSB)
7	ext. address
8	address (MSB)
9	address
10	address
11	address (LSB)
12	data length (MSB)
13	data length
14	data length (LSB)
15	header CRC
16	data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

	Title:	**Unverified Write – Instruction Field**
FEE-DPU-IF-586	For a write request to the critical configuration area, the RMAP instruction shall be 0x6C. Regarding AD22 this means: - 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:	**Unverified Write – Length Field**
FEE-DPU-IF-587	The data length of a unverified write request shall be a multiple of 4 bytes. The data-length shall be less than or equal to 16,384 Bytes.	

	Title:	**RMAP Read Request**
FEE-DPU-IF-588	A RMAP read-request shall be used by the DPU to get the content of any FEE-register.	

Figure: Request-packet for read access



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**
FEE-DPU-IF-589	For a read request to the critical configuration area, the RMAP instruction shall be 0x4C. Regarding AD22 this means: - 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 Read – Data Length Field**
FEE-DPU-IF-590	The data length of a read request shall be a multiple of 4 bytes. The data-length shall be less than or equal to 16,384 Bytes.

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

	Title: **RMAP Request – Initiator Address Field**
FEE-DPU-IF-592	The initiator address field in a RMAP request shall be 0x50.

	Title: **RMAP Request – Transaction ID Field**
FEE-DPU-IF-826	The DPU shall increment the transaction ID for each RMAP request.

	Title: **RMAP Request – Address Field**
FEE-DPU-IF-827	The address field in a RMAP request shall contain the FEE register address. The address shall be aligned to 32-bit, i.e. bits 0 and 1 are zero. The extended address will not be used and shall be zero.

	Title: **RMAP Request – Header and Data CRC Field**
FEE-DPU-IF-828	The header and data CRC of a RMAP request shall be calculated by DPU and FEE as described in AD03.

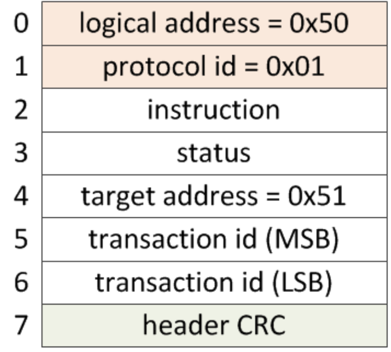
	Title: **RMAP Request – Header CRC Check**

FEE-DPU-IF-829	The FEE shall check the header CRC for a RMAP request.

	Title:	**RMAP Request – Data CRC Check**
FEE-DPU-IF-830	The FEE shall check the data CRC for a RMAP write request.	

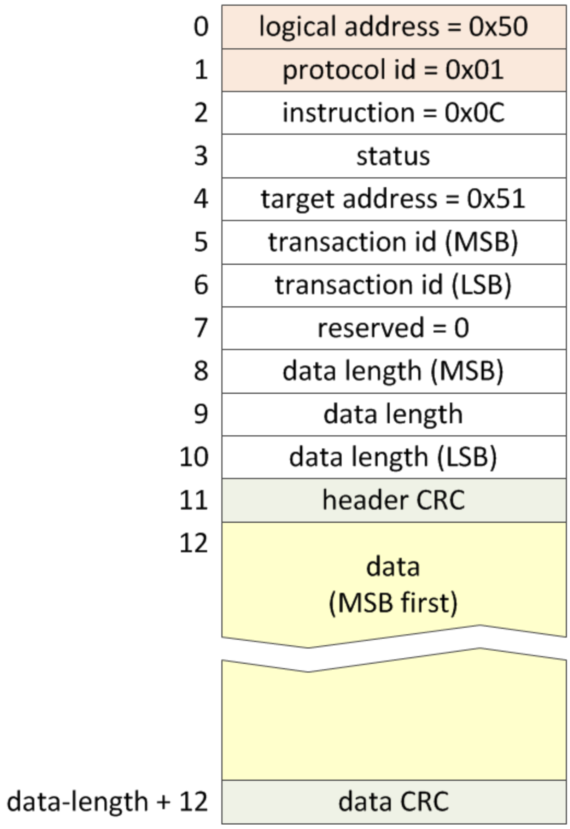
	Title:	**RMAP Reply**
FEE-DPU-IF-832	The FEE shall send a RMAP-reply to any RMAP request to Spacewire address 0x51.	

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

Figure: RMAP-reply packet to a read-request



0	logical address = 0x50
1	protocol id = 0x01
2	instruction = 0x0C
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**
FEE-DPU-IF-834	The logical-address for the Spacewire-routing shall contain the initiator address from the request-packet.	

	Title:	**RMAP Reply – Instruction Field**

FEE-DPU-IF-835	The instruction field of the RMAP-reply shall have 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**
FEE-DPU-IF-836	The status-field in the RMAP-reply shall contain 0 if the command execution was successful. An error in the RMAP request execution shall cause a non-zero status-field according to AD22. The following status-codes shall be supported by the FEE: 0-7, 12.

	Title: **RMAP Reply – Target Field**
FEE-DPU-IF-837	The target address field in the RMAP-reply shall contain 0x51.

	Title: **RMAP Reply – Transaction ID Field**
FEE-DPU-IF-838	The transaction ID field in the RMAP-reply shall contain of copy of transaction ID in the corresponding RMAP request.

	Title: **RMAP Read Reply – Data Length Field**
FEE-DPU-IF-839	The data-length field in the RMAP-reply shall contain of copy of data-length in the corresponding RMAP read-request.

	Title: **RMAP Reply – Header and Data CRC Field**
FEE-DPU-IF-840	The header and data CRC of a RMAP reply shall be calculated by DPU and FEE as described in AD03.

	Title: **RMAP Reply – Header CRC Check**
FEE-DPU-IF-868	The DPU shall check the header CRC for a RMAP reply.

	Title: **RMAP Reply – Data CRC Check**
FEE-DPU-IF-841	The DPU shall check the data CRC for a RMAP read reply.

	Title: **RMAP Reply - Transaction ID Check**
FEE-DPU-IF-907	The DPU shall try to match the RMAP-reply to the RMAP-request by checking the transaction-ID.

	Title: **RMAP Reply Period**
FEE-DPU-IF-842	The FEE shall send a reply for every RMAP request inside 10 milliseconds after the end of the request-packet.

	Title: **RMAP-FDIR – Request Timeout**
FEE-DPU-IF-843	If the DPU does not receive a reply to a RMAP request inside 100 milliseconds, it shall repeat the request.

	Title: **RMAP Write to Illegal Addresses**
FEE-DPU-IF-844	RMAP write requests to unused addresses or across memory borders shall be reported by the FEE with status-code 1.

Note: Because the address does not exists, the write itself will be ignored by the FEE.

	Title: **RMAP Read from Unused Addresses**

FEE-DPU-IF-845	RMAP read requests to unused addresses shall be reported by the FEE as successful (status = 0) and shall return 0 as data.

	Title:	**Open RMAP requests**
FEE-DPU-IF-936	Before a RMAP request is send by the DPU, all other RMAP request shall be closed (i.e. replied).	

	Title:	**RMAP-FDIR – Invalid Header CRC in Request**
FEE-DPU-IF-846	RMAP request, where the check of the header CRC fails, shall be ignored by the FEE.	

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

	Title:	**RMAP-FDIR – Invalid Data CRC in Request**
FEE-DPU-IF-847	If the data-CRC-check of a RMAP write-request fails, the FEE shall reply with status-code 4.	

	Title:	**RMAP-FDIR – Invalid Key**
FEE-DPU-IF-848	RMAP requests with a key other than 0xD1 shall be replied by the FEE with status-code 3.	

	Title:	**RMAP-FDIR – Invalid Logical FEE-Address**
FEE-DPU-IF-849	RMAP requests to a logical address other than 0x51 shall be replied by the FEE with status-code 12.	

	Title:	**RMAP-FDIR – Invalid Protocol ID**
FEE-DPU-IF-850	Spacewire packets received by the FEE with a protocol-ID other than 0x01 shall be ignored.	

Note: No reply shall be send to the DPU on an invalid protocol-ID.

	Title:	**RMAP-FDIR – Invalid or Unsupported Instruction**
FEE-DPU-IF-852	If a RMAP request contains an instruction other than 0x7C, 0x6C or 0x4C, the FEE shall reply with status-code 2.	

	Title:	**RMAP-FDIR – Invalid Verified Write Data Length**
FEE-DPU-IF-853	If a RMAP verified-write-request contains a data-length other than 4, the FEE shall reply with status-code 1.	

	Title:	**RMAP-FDIR – Data Length to Large**
FEE-DPU-IF-854	If a RMAP-request contains a data-length larger than 16,384 Bytes, the FEE shall reply with status-code 6.	

	Title:	**RMAP-FDIR – Invalid Length Alignment**
FEE-DPU-IF-855	If a RMAP-request contains a data-length, which is not a multiple of 4, the FEE shall reply with status-code 1.	

	Title:	**RMAP-FDIR – Invalid Header CRC in Reply**
FEE-DPU-IF-858	The DPU shall ignore RMAP-reply packets, where the header CRC-check failed.	

	Title:	**RMAP-FDIR – Invalid Data CRC in Reply**
FEE-DPU-IF-859	The DPU shall repeat the RMAP read-request, when the header data CRC-check failed.	

	Title: **RMAP-FDIR – Invalid Target Address**
FEE-DPU-IF-860	The DPU shall repeat the RMAP request, when the target address in the RMAP reply is invalid.

	Title: **RMAP-FDIR – Invalid Status**
FEE-DPU-IF-861	The DPU shall repeat the RMAP request, when the status the RMAP reply is not zero.

	Title: **RMAP-FDIR – Invalid Transaction ID**
FEE-DPU-IF-862	If the transaction-ID of a RMAP reply does not match any open request, the DPU shall repeat the RMAP request.

	Title: **RMAP-FDIR – Request Repeats**
FEE-DPU-IF-863	The DPU shall retry a RMAP-request after a fail. The time between retries shall be configurable in a range of 0-10 seconds. The maximum number of retries shall be configurable in a range of 0..31.

	Title: **RMAP-FDIR – DPU Error Counter**
FEE-DPU-IF-864	The DPU shall contain an error-counter, which shall be incremented on any kind of RMAP error (time-out, reply-status not 0, invalid reply).

	Title: **RMAP-FDIR – DPU Error Report**
FEE-DPU-IF-865	The DPU shall report every RMAP error in the housekeeping-data with an unambiguous code.

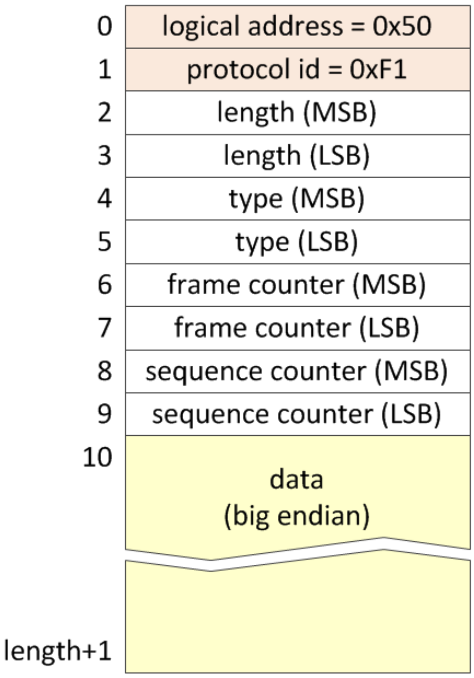
	Title: **RMAP-FDIR - Multiple Errors**
FEE-DPU-IF-935	If multiple errors occure in a RMAP request, only one error shall be reported.

8 Data Interface

	Title: **Data Packet Format**
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

Data and HK Packet



0	logical address = 0x50
1	protocol id = 0xF1
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+1	

	Title: **Data Packet Header**
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**
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: **Windowing Packet Length (N-FEE)**
FEE-DPU-IF-873	In windowing-mode the N-FEE data packets shall have a length of 18250 bytes. Only the last packet of the current image may be shorter.

	Title: **Windowing Packet Length (F-FEE)**
FEE-DPU-IF-874	In windowiwng-mode the F-FEE data packets shall have a length of 128 bytes. Only the last packet of the current image may be shorter.

Note: Because the F-FEE does not have an external memory and the block-RAMs inside the FPGA are limited the send buffer cannot hold more than 128 Bytes.

	Title: **Data Packet Length in Calibration-Mode**
FEE-DPU-IF-875	In windowing-mode the N-FEE data packets shall have a length of 18248 bytes. Only the last packet of the current image may be shorter.

Note: The length of 18250 was choosen to hold 4 complete lines in a packet (10 byte header + 2 bytes * 4560 pixels * 4 lines).

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

	Title: **Spacewire Protocol ID for FEE-Data**
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**
FEE-DPU-IF-877	Bytes 2 and 3 of the data-packet-header contain the data-length in bytes.

	Title: **Data Packet Field: Type**
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 = calibration mode, 1= calibration 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.

	Title: **Data Packet Field: Frame Counter**
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: Sequence Counter**
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 a global counter for all kinds of data packets.

Note: Because HK packet is send before the image data, the HK sequence-counter will be 0 and the sequence-counter of first image-packet will be 1.

	Title: **Sequence Counter Consistency-Check**
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.

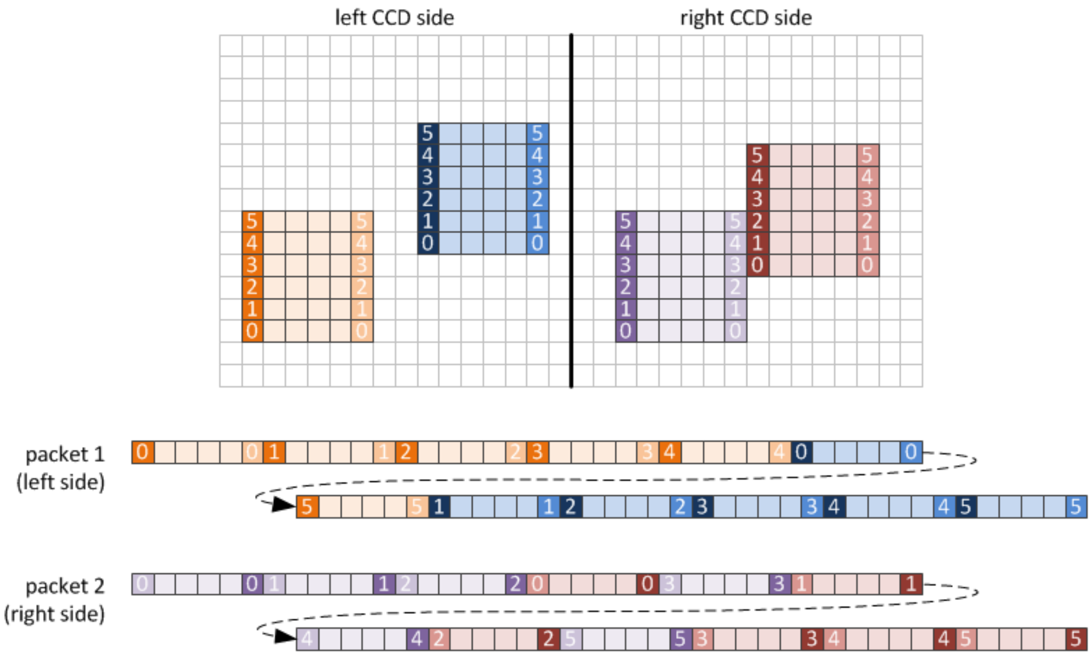
	Title:	**Data Packet Field: Data**
FEE-DPU-IF-883	Depending on the type-field, the data-field contains either image-data or housekeeping-data.	

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

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

	Title:	**Data Read-Out Order**
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**
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**
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**
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**
FEE-DPU-IF-890	The HK packet shall be send before the image data.	

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

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

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

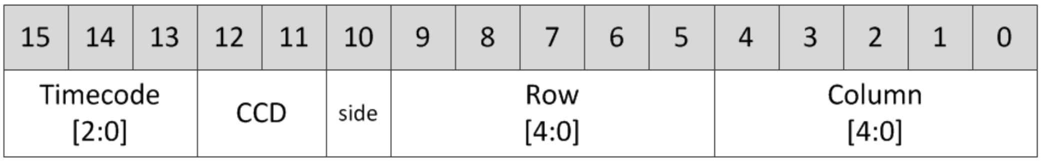
9 Pattern Generation

	Title:	**Pattern Generation**
FEE-DPU-IF-896	If the FEE is calibration-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 calibration mode.	

	Title:	**Pattern Structure**
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 calibration-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