DOCUMENT

Packet Structure- Interface Control Document

FIRST/Planck Project

prepared by	S. Thürey (SCI-PXI)
approved by	Th. Passvogel (SCI-PT)
reference	SCI-PT-ICD-7527
issue	1
revision	0
date of issue	1 Septomber 2000

European Space Agency
Agence spatiale européenne
ESTEC
Keplerlaan 1 - NL 2201 AZ Noordwijk ZH - The Netherlands
Tel (31) 71 565 3473 - Fax (31) 71 565 5244 - http://sci.esa.int/first -- http://sci.esa.int/planck

CHANGE LOG

Date	Issue No.	Rev. No.	Pages Affected	Description/Authority	CR No.
1-9-2000	1	0	All	New Issue	---

Revisions are normally indicated by a vertical bar at the outside border.

DISTRIBUTION LIST

Recipient	Organisation	Number of Copies
M. Anderegg	SCI-PT	1
H. Barre	SCI/PXS	1
B. Collaudin	TOS-MCT	
A. Elfving	SCI-PXS	1
P. Estaria	SCI-PT	1
B. Guillaume	SCI/PT	1
A. Heske	SCI-PS	1
M. von Hoegen	SCI-PT	1
B. Jackson	SCI-PXS	1
P. de Maagt	TOS-EEA	
R. Morgan-Owen	TOS-ETT	1
Th. Paßvogel	SCI-PT	1
M. Ranne	SCI-PMO	
S. Thürey	SCI-PXI	1
F. Vandenbussche	SCI-PT	1
F. Wechsler	SCI-PXI	1
FIRST Project File		1
SA-DMS		1
G. Pilbratt	SCI-SA	1
J. Tauber	SCI-SA	1
S. Dodsworth	TOS-OFC	1
M. Hechler	TOS-OFA	1
B. Smeds	TOS-ONS	1
HIFI Team		
Th. de Graauw	SRON. Groningen.	
H. J. M. Aarts	SRON. Utrecht.	2
PACS Team		
PACS Project Office	MPE, Garching	2
SPIRE Team		
M. Griffin	QMW, London	1
K. King	RAL, Oxfordshire	1
HFI Team		
J.-L. Puget	IAS, Orsay	1
J. Charra	IAS, Orsay	1
LFI Team		
N. Mandolesi	TeSRE-CNR, Bologna	
C. Butler	TeSRE-CNR, Bologna	2
R. Orfei	CNR-IFSI, Roma	1

TABLE OF CONTENTS

1 INTRODUCTION AND SCOPE

1.1 Scope

The ESA Packet Telemetry and Telecommand Standards (**[AD-1]** and **[AD-2]**) address the transport of telemetry and Telecommand data between user applications on the ground and user applications on-board of a satellite, and the intermediate transfer of this data through the different elements of the ground and space segments.

The Packet Structure ICD serves to complement and extend the Packet Telemetry and Telecommand Standards by defining all the mission specific details of the **data transfer between ground and on-board applications** (i.e. on-board units, subsystems, and instruments, and their counterparts on ground).

For the **Application Layer Interface** this document describes procedures and **the data structures for Telemetry and Telecommand Packets** to be implemented for support of the operational requirements defined in the FIRST / Planck Operations Interface Requirements Document **[AD-3]**. Being tailored especially for FIRST / Planck, this document supersedes the Packet Utilisation Standard (PUS) **[RD-1]** from which it is derived.

Protocol layers below the Application Layer, which serve for transferring TC- and TM-Packets on a **physical medium between on-board units**, also have to be defined in order to be compliant with general and mission-specific requirements. Therefore, this document specifies in **Appendix 9** a Satellite Data Bus Protocol, which provides the necessary definitions for controlling the on-board data transfer within the **Physical, Data Link Layer and Transfer Layer.**

The FIRST / Planck space and ground segment will support all and only the data structures and services defined in this document. The document will be agreed between the ESA FIRST / Planck Project Manager, the appointed ESOC representative and the Prime Contractor. Upon approval, the document will be controlled by the ESA FIRST / Planck Project Office to whom any updates or changes to any parameters contained in it shall be submitted.

1.2 Operations Scenario

1.2.1 Nominal Mission Operations

The mission operations of the FIRST / Planck spacecraft and their payloads will be conducted under control of the Mission Operations Centre (MOC, ground) at the European Space Operations Centre (ESOC). Throughout the complete mission duration (from launch up to the end of mission, when ground contact to the spacecraft/payload is terminated), facilities and services will be provided to the Science Centre for planning and execution of astronomical observations, and provision of the necessary data sets.

Interaction with the spacecraft will be by monitoring and analysis of telemetered data and the uplink of commands to effect the necessary operations. Most **Telecommands will be stored on board** for later execution at a defined time, via the On-board Operation Scheduling Service, **others** may be intended for **immediate execution**. In both cases, it may be necessary to control subsystem and experiment equipment via on-board applications, which may interact with physical interfaces and/or serve for internal data processing.

In the order of increasing complexity **Task** or **Function Management** Services or **On-Board Control Procedures** (OBCPs) can be used for these purposes. The OBCPs can constitute autonomous on-board control loops, which make use of the exchange of TM/TC Packets, can affect more than one unit, and may be active for a considerable period of time. They shall be kept simple and reduced to the **essential minimum**.

Telemetry and Telecommand services are provided in support of **nominal** management functions, including:

- sampling and processing of periodic housekeeping data,
- direct commanding of hardware interfaces,
- control of, and communication with, on-board tasks, functions and on-board control procedures,
- control of the on-board mission time line,
- recording to and retrieving data from an on-board mass memory,
- monitoring of and reacting on asynchronous on-board events.

All Telecommands must be appropriately verified by Telemetry Reports at acceptance and execution. Whereas acknowledge of acceptance is always done via an Event TM Packet of a special type, Telecommand execution progress and completion can be reported by (a combination of) Event Packets and/or periodic housekeeping data, depending on the exact nature of the executed function.

Telemetry data will be required in order to verify the execution of all mission operations, and will also be required for:

- routine on-ground status and health monitoring of the subsystems and the experiments;
- reporting to the ground any anomalous events detected on-board and any actions taken autonomously by the on-board systems;
- performance evaluation on the ground for the purposes of long-term trend analysis and feedback into the mission planning cycle.

1.2.2 Contingency Operations

In the event of unforeseen on-board events, on-ground actions will be necessary to investigate and correct anomalies utilising the **available telemetry and command functionality**.

Several TM / TC packet services in support of unit-specific data structures are implemented in support of contingency operations:

- activation or modification of test or diagnostic operational modes and associated TM packet structures,
- modification of the on-board operations schedule,
- modification of on-board control procedures,

In **exceptional cases**, it may be necessary to modify on-board parameters or software in order to compensate for on-board failures or anomalous performance. This may be done by dumping, checking, and loading of on-board memories.

1.2.3 Packet Distribution

The following Telemetry and Telecommand Packet categories exist:

- those generated on the ground and up-linked to the spacecraft for immediate distribution or intermediate storage,
- those generated by on-board applications and down-linked to the ground,

- those generated on-board and routed to other on-board applications (and to the ground in all cases).

These packet categories are to be routed to or from on-board end-users which are capable of handling the services associated with TM / TC Packet data structures (so-called packet end users). The routing of packet data takes place under control of the Command and Data Management System (CDMS) of the satellite, which is in charge of initialisation, timing, and prioritising of any on-board packet data transfer.

The CDMS is also acting as the controlling subsystem for the on-board serial data bus, which serves for physically connecting units and routing data from or to instruments, other spacecraft units, or subsystems. The protocol, which serves for transferring packet data and other data (e.g. control information), does not impose restrictions on packet data structures, and the associated service acts independently from packet services. Details of this protocol are specified in Appendix 9 of this document.

The management of the serial data bus incorporates certain Failure Detection, Isolation, and Recovery (FDIR) functions, which provide for a reliable transfer of packet data. They are also defined in Appendix 9. These FDIR functions are executed and reported by the CDMS; all other packet end-users are required to generate the appropriate event packets in case problems with the exchange of data can be detected.

1.3 APPLICABLE DOCUMENTS

1. Packet Telemetry Standard, PSS-04-106, Issue 1, January 1988
2. Packet Telecommand Standard, PSS-04-107, Issue 2, April 1992
3. FIRST / Planck Operations Interface Requirements Document (F/P-OIRD), SCI-PT-RS-07360, Issue 1

1.4 REFERENCE DOCUMENTS

1. Packet Utilisation Standard (PUS), ESA PSS-07-101, Issue 1 (ECSS-E-70/41, Draft 04)
2. CCSDS Packet Telemetry, CCSDS 102.0-B-4, November 1995
3. MIL-Std.-1553 B, Digital Internal Time Division Command/Response Multiplex Data Bus, Issue Notice 2, 8 September 1986

2 PACKET SERVICES

The Packet Services listed in

Table 2-1 below are available on FIRST/Planck for users and are specified in detail within the following chapters.

Service Type	Service Name	Applicability for Instruments and S/Ss other than CDMS
1	Telecommanding and TC- Verification	Mandatory
2	Device Command Distribution Service	
3	Housekeeping and Diagnostic Data Reporting	Mandatory
4	not used	
5	Event Reporting	Mandatory
6	Memory Management	Mandatory
7	Task Management	Optional
8	Function Management	Optional
9	Time Management Service	Mandatory
10	not used	
11	On-board Operations Scheduling Service	
12	On-board Monitoring Service	Optional
13	not used	
14	Packet Transmission Control Service	Mandatory
15	On-board Storage and Retrieval Service	
16	On-board Traffic Management	
17	Test Service	Mandatory
18	On-board Control Procedure Service	Optional
19	Event/Action Service	
20	Information Distribution CDMS – Users TBC	Optional
21	Science Data Transfer Service	Mandatory (for Instruments only)
22	Context Saving TBC	Optional

Table 2-1: Packet Services specified within this document

3 TELECOMMAND STRUCTURE

3.1 Telecommand Source Packets

All Telecommand source packets must conform to the structure defined in **[AD-2]** and shown in Figure 3.1-1 below.

<table>
<tr><th colspan="6">PACKET HEADER (48 bits)</th><th colspan="3">PACKET DATA FIELD
(VARIABLE)</th></tr>
<tr><td colspan="4">PACKET ID</td><td colspan="2">PACKET SEQUENCE CONTROL</td><td rowspan="3">PACKET LENGTH</td><td rowspan="3">DATA FIELD HEADER</td><td rowspan="3">APPLIC-ATION DATA</td><td rowspan="3">PACKET ERROR CONTROL</td></tr>
<tr><td>Version Number</td><td>Type</td><td>Data Field Header Flag</td><td>APID</td><td>Sequence Flags</td><td>Sequence Count</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>11</td><td>2</td><td>14</td></tr>
<tr><td colspan="4">16 bits</td><td colspan="2">16 bits</td><td>16 bits</td><td>32 bits</td><td>N x 16 bits</td><td>16 bits</td></tr>
</table>

Figure 3.1-1 Telecommand Packet Fields

3.1.1 Packet Header

3.1.1.1 Packet ID

3.1.1.1.1.1 Version Number:
The Version Number must be set to ‘000’ $_{\text{BIN}}$ for all commands.

3.1.1.1.1.2 Type:
This bit distinguishes between Telecommand packets and telemetry source packets. For Telecommand packets, the type = 1.

3.1.1.1.1.3 Data Field Header Flag:
This indicates the presence of the Data Field Header when set to 1.

All commands except CPDU commands, TC Packet Type/Subtype (2,3), shall have a data field header.

3.1.1.1.1.4 Application Process ID:

The Application Process ID (APID) defines the application or unit which the Telecommand **is addressed to.**

The choice of Application Process ID values across the spacecraft subsystems and experiments are given in Appendix 3.

3.1.1.2 Packet Sequence Control

3.1.1.2.1.1 Sequence Flags:

These 2 bits shall be set to “11”, all Telecommands shall be “stand-alone” packets.

3.1.1.2.1.2 Sequence Count: (14 bits):

This field is provided to identify a particular Telecommand packet so that it can be traced within the end-to-end Telecommand system. The field is divided into two parts as follows:

Source part (3 most significant bits) identifies the generator or source of a certain command as follows:

000 = Ground, all sources (maintained by ground)
001 = Mission Time Line (maintained by ground)
010 = CDMS, all sources except mission time line (maintained by CDMS on-board)
011 = AOCS, (maintained by AOCMS on-board) TBC
100-111 = Spare

Sequence part (11 bits) shall be used to represent the actual Sequence Count. The Sequence Count is maintained by the Telecommand source for each Application Process ID. The sequence count shall be incremented by 1 whenever a command is generated with that Application Process ID. The counter wraps around from “full-scale” to zero.

When an acknowledgement of a TC-packet is required (see “Ack” field in the data field header below), it is mandatory that the full Sequence Control field is included in the telemetry acknowledge packet as the identifier of the Telecommand packet being acknowledged.

No check is to be performed by the addressed application regarding the monotony of the sequence counter, **the application shall accept commands regardless of the sequence counter**.

3.1.1.3 Packet Length

The Packet Length field specifies the number of octets contained within the Packet Data Field. The number is an unsigned integer “C” where

3.1.1.3.1.1 C = (Number of octets in Packet Data Field) – 1
The maximum length of a Telecommand Packet Data Field is **242 octets** (i.e. C <= 241). The overall TC Packet including the Packet Header is 6 octets longer, therefore the **maximum TC Packet length is 248 octets.**

The Packet Length shall be an integer number of 16 bit words, as a result C will always be a odd number (of octets).

3.1.2 Packet Data Field

3.1.2.1 Data Field Header

The data field header is preceded by the packet header and followed by application data and error control in the Telecommand packet, refer to figure 4.1.2-1. The data field header is defined as follows:

CCSDS Secondary Header Flag	TC Packet PUS Version Number	Ack	Packet Type	Packet Subtype	Spare
Boolean (1 bit)	**Enumerated (3 bits)**	**Enumerated (4 bits)**	**Enumerated (8 bits)**	**Enumerated (8 bits)**	**(8 bits)**

CCSDS Secondary Header Flag:

This bit shall be set to zero to indicate that the PUS Data Field Header is a "non-CCSDS defined Secondary Header".

TC Packet PUS Version Number:

ONLY ONE PUS VERSION NUMBER IS PERMITTED: VERSION 0 (VALUE = 0).

Ack:

This field is used to indicate which acknowledgements, in the form of Telecommand verification packets, are required to notify acceptance and to verify execution of this Telecommand packet. This relates only to acknowledgement of successful acceptance and execution, since failure reports shall be generated by default.

The bit settings shall be as follows:

---1 (bit 3 of the Ack field set): **mandatory**, acknowledge acceptance of the packet by the Application Process

--1- (bit 2 of the Ack field set): acknowledge start of execution

-1-- (bit 1 of the Ack field set): acknowledge progress of execution

1--- (bit 0 of the Ack field set): acknowledge completion of execution.

Packet Type:

This indicates the Service to which this packet relates.

Packet Subtype:

Together with the Packet Type, the Subtype uniquely identifies the nature of the Service Request constituted by this Telecommand packet.

The definition of Packet Type and Subtype is unique across all Application Processes.

Spare:

Spare bits are introduced in order to make up an integral octet. These spare bits shall be set to zero.

3.1.2.2 Application Data

The Telecommand application data constitute the data element of the Telecommand to be used by the application.

3.1.2.3 Packet Error Control (PEC) (16 bits)

The purpose of the mandatory Packet Error Control field is to transport an error detection code that shall be used by the receiving Application Process to verify the integrity of the complete Telecommand Packet. The type of the PEC is fixed for the complete mission for each Application Process and is defined in Appendix 4.

4 TELEMETRY STRUCTURE

4.1 Telemetry Source Packet

All Telemetry Source Packets must conform to the structure defined in **[AD-1]** and shown in Figure 4.1-1 below.

SOURCE PACKET HEADER (48 bits)							PACKET DATA FIELD (VARIABLE)		
PACKET ID				PACKET SEQUENCE CONTROL		PACKET LENGTH	DATA FIELD HEADER	SOURCE DATA	PACKET ERROR CONTROL
Version Number	Type	Data Field Header Flag	Application Process ID	Segmentation Flags	Source Sequence Count				
3	1	1	11	2	14				
16 bits				16 bits		16 bits	80 bits	N x 16 bits	16 bits

Figure 4.1-1 Telemetry Source Packet Fields

4.1.1 Source Packet Header

4.1.1.1 Packet ID

4.1.1.1.1.1 Version Number:
The Version Number must be set to ’000’$_{\text{BIN}}$ (The specification in this document is consistent with [RD-2] and supersedes [AD-1]) for all telemetry issued on-board. The ground segment shall reject with an alarm any packet received with a version number other than zero.

4.1.1.1.1.2 Type:
For Telemetry Source Packets, the type must be set to zero.

4.1.1.1.1.3 Data Field Header Flag:

This indicates the presence or absence of a Data Field Header and must be set to 1 except for Time Packets and for Idle Packets where it is set to 0 (see Service 9, and Appendix 7).

4.1.1.1.1.4 Application Process ID (APID):

The Application Process ID uniquely identifies the **on-board source** of the packet.

For Telemetry Packets the APIDs have an internal structure that allows an allocation of each TM-Packet to one of three major service categories. The choice of Application Process ID values across the spacecraft subsystems and experiments are given in Appendix 3.

Two Application Process ID’s have been reserved for special purposes, namely the Standard

Spacecraft Time Source Packet and the Idle Packet. Their use and data structure are provided in Appendices 7 and 8 respectively. Additionally, a range of APIDs is allocated exclusively for EGSE-related messages. These APIDs shall not be used for any on-board TM/TC-Packets.

4.1.1.2 Packet Sequence Control

4.1.1.2.1.1 Segmentation [Grouping] Flags:

These two bits shall be set to ‘11’$_{\mathrm{BIN,}}$ indicating “no segmentation”.

An exception to the above rule is made for Science Data packets TM(21,x). These packets may use the segmentation flags to implement packet grouping as defined below.

segmentation flags = 01: first source packet of a group
segmentation flags = 00: continuation source packet of a group
segmentation flags = 10: last source packet of a group
segmentation flags = **11: a self standing source packet** not belonging to a group.

The packet order within a group is given by the source sequence count field.

Note that the onboard data management system and the ESA ground data system does not provide any special support for the grouping of source packets. All packets belonging to a group are treated as self standing data units. It shall be the responsibility of the user to support the packet grouping in the onboard instrument and in the ground data processing equipment.

4.1.1.2.1.2 Source Sequence Count:

A separate source sequence count shall be maintained for each Application Process ID and shall be incremented by 1 whenever the source (APID) releases a packet. Therefore the counter corresponds to the order of release of packets by the source and enables the ground to detect missing packets.

4.1.1.3 Packet Length

The Packet Length field specifies the number of octets contained within the Packet Data Field, including the Data Field Header. The number is an unsigned integer “C” where

4.1.1.3.1.1 C = (Number of octets in Packet Data Field) - 1
For FIRST / Planck the maximum length of a Telemetry Source Packet Data Field is **1018 octets**, i.e. the maximum value for C is 1017.

It should be noted that the actual length of the entire Telemetry Source Packet, including the Source Packet Header, is 6 octets longer. Therefore the **maximum TM Packet length is 1024 octets.**

The Packet Length shall be an integer number of 16 bit words, as a result C will always be a odd number (of octets).

4.1.2 Packet Data Field

4.1.2.1 Data Field Header

The content of the Data Field Header depends on the nature of the Telemetry Reports defined in the remainder of this document, however all data field headers have the same basic structure, as follows:

Spare	TM Source Packet PUS Version Number	Spare	Packet Type	Packet Subtype	Spare	Time
Bitstring (1 bit)	Enumerated (3 bits)	Bitsring (4 bits)	Enumerated (8 bits)	Enumerated (8 bits)	Bitsring (8 bits)	(48 bits)

Spare:

To maintain symmetry with the Telecommand packet Data Field Header, this bit is reserved and shall be set to zero

TM Source Packet PUS Version Number:

ONLY ONE PUS VERSION NUMBER IS PERMITTED: VERSION 0 (VALUE = 0).

Spare:

Spare bits are placed in order to make up an integral octet. These spare bits shall be set to zero.

Packet Type:

This indicates the Service to which this telemetry source packet relates.

Packet Subtype:

Together with the Packet Type, the Subtype uniquely identifies the nature of the Service constituted by this telemetry source packet.

The definition of Packet Type and Subtype is unique across all Application Processes

Spare:

Spare bits are introduced in order to make up an integral octet. These spare bits shall be set to zero.

Time:

This field represents the on-board reference time of the packet, referenced to TAI, expressed in CUC. Details of the time field are given in appendix 6.

The relationship of the time information to packet data generation or packet completion shall be fixed and defined per packet type/subtype of each application.

4.1.2.2 Source Data (Variable)

The telemetry source data constitutes the data element of the TM Packet

4.1.2.3 Packet Error Control (PEC) (16 bits)

The Packet Error Control field shall transport an error detection code that can be used by the ground to verify the integrity of the complete telemetry source packet. The presence of the PEC and its type is fixed for the complete mission for each Application and defined in Appendix 4.

5 Packet Data Field Structures

5.1 Service Type 1: Telecommand Verification Service

Telecommand Verification Report Packets are a specific type of Event Packets, and generated by on-board applications after reception and potentially execution of a Telecommand. The APID of the TC Verification Report shall be identical to the APID of the Telecommand packet being acknowledged.

5.1.1 Telecommand Packet Data Field Structure:

not applicable

5.1.2 Telemetry Packet Data Field Structure:

5.1.2.1 Telecommand Acceptance

For **all** Telecommands, which pass the acceptance checks when received by the executing unit or application, a TC-Acceptance Report Packet shall be generated immediately, i.e. in less than 2 seconds after reception.

The reports of acceptance of a Telecommand Packet are as follows:

Telecommand Acceptance Report - Success (1,1)

Telemetry Source Packet, Source Data:

Telecommand Packet ID	Packet Sequence Control
2 octets	2 octets

Telecommand Packet ID:

This is a copy of the corresponding field from the packet header of the Telecommand to which this verification packet relates i.e. the APID (and the most significant 5 bits of this field).

Packet Sequence Control:

This is a copy of the corresponding fields from the packet header of the Telecommand to which this verification packet relates.

Note:

During longer periods of autonomous operation of FIRST and Planck Telecommands may be generated or released from several (sometimes independent) sources. An instrument for example may receive commands from the Mission Timeline, On-board Control Procedures, Failure Detection, Isolation and Recovery Procedures, and potentially instrument-internal OBCPs. It may additionally be the case that the actual value of parameters, which are handed over to an Application, cannot be unambiguously determined on ground.

As only all **Telemetry** packets are recorded on-board and downlinked, the actual sequence of on-board generated Telecommands, and their contents, is unknown on ground with the current definition of the PS-ICD. The Telecommand Verification reports, as derived from the PUS [RD 1], are insufficient for making all needed information available.

In the next issue of this document the adequate TM packet structure for this case, TBD, will be added.

Telecommand Acceptance Report - Failure (1,2)

In case of rejection of **any** Telecommand by a unit or application an event-packet with the data field below shall be generated:

Telemetry Source Packet, Source Data:

Telecommand Packet ID	Packet Sequence Control	Code	Parameters
2 octets	2 octets	Enumerated, 1 octet	Any
			<------- Optional ------->

Telecommand Packet ID:

This is a copy of the corresponding field from the packet header of the Telecommand to which this verification packet relates i.e. the APID (and the most significant 5 bits of this field).

.

Packet Sequence Control:

This is a copy of the corresponding fields from the packet header of the Telecommand to which this verification packet relates.

Code:

The following standard reasons for failure of acceptance of a Telecommand shall be reported:

0 = illegal APID

1 = incomplete or invalid length packet;

2 = incorrect checksum;

3 = illegal packet Type;

4 = illegal packet Subtype;

16 = illegal or inconsistent Application Data

16 to 255 = application-specific failure of acceptance, to be defined in the User Manual of the unit or subsystem involved, subject to approval by ESA..

Parameters:

For the standard reasons above the illegal or incorrect parameter shall be reported.

For application-specific command rejections the parameter field is to be defined by users and provides complementary information relating to the particular value of the code field.

In all cases the code field plus the parameter field shall have a length corresponding to one or several 16-bit words. **The maximum length for the code plus parameter fields shall not exceed 40 octets, for a total length of the TC packet of less than 64 octets.**

5.1.2.2 Telecommand Execution Started:

The **nominal case** for Telecommands shall be that they are executed **immediately** after reception. The timing of Telecommand execution is controlled by the Mission Timeline on-board or directly from ground.

For Telecommands, which cannot fulfil this criterion, a TC Execution Started Report may be generated, if the (start of) execution is not reported by **periodic housekeeping.** The data fields for reports of start of execution of a Telecommand packet are as follows:

Telecommand Execution Report - Started (1,3)

Telemetry Source Packet, Source Data:

Same as for Type/Subtype (1,1)

5.1.2.3 Telecommand Execution Progress:

Only in case the execution of a Telecommand consists of a significant number of sub-steps and the overall execution takes **several tens of seconds or more**, it may be appropriate to report the successful completion of these steps by a Telecommand of type (1,5).

The data field for reports of progress of execution of a Telecommand packet are as follows:

Telecommand Execution Report - Progress (1,5)

Telemetry Source Packet, Source Data:

Telecommand Packet ID	Packet Sequence Control	Step Number
2 octets	2 octets	Enumerated, 2 octets

Telecommand Packet ID:

This is a copy of the corresponding field from the packet header of the Telecommand to which this verification packet relates, i.e. the APID (and the most significant 5 bits of this field).

Packet Sequence Control:

This is a copy of the corresponding fields from the packet header of the Telecommand to which this verification packet relates.

Step Number:

This indicates the intermediate step number of the Telecommand execution profile whose execution has been completed. The values it can take are Telecommand specific.

5.1.2.4 Telecommand Execution Completed

If the successful overall completion of execution of a Telecommand can be detected by an application, and if the completed execution is not reported satisfactorily by other means (e.g. periodic housekeeping data), a Telecommand Execution Report - Success, type (1,7), shall be generated.

The reports of completion of execution of a Telecommand packet are as follows:

Telecommand Execution Report - Completed (1,7)

Telemetry Source Packet, Source Data:

Same as for Type/Subtype (1,1)

5.1.2.5 Telecommand Execution Failure

For **all** other cases, i.e. for not started, unsuccessfully executed, or otherwise aborted TC executions an event-message of type (1,8) can be generated, if the unit or application involved is capable of doing this.

Telecommand Execution Report - Failure (1,8)

Telemetry Source Packet, Source Data:

Same as for Type/Subtype (1,2).

Execution Failure Codes and Parameters: application–specific, to be defined in the User Manual of the unit or subsystem involved.

5.2 Service Type 2: Device Command Distribution

The Telecommands of service type 2 are foreseen for the control of units which are equipped with hardware interfaces for immediate execution of these commands. Completion of execution for these functions can only be indicated by periodic housekeeping in several cases.

5.2.1 Telecommand Packet Data Field Structure:

5.2.2 Distributing Pulse Commands

The request for the distribution of pulse command(s) by means of a Telecommand packet is:

Distribute Pulse Commands (2,1)

Telecommand Packet, Application Data:

N	Address
Unsigned Integer	**Unsigned Integer**
Optional, TBC	<--- Repeated N times --->

N:

The number of Pulse Commands which follow (N > 0, N_max. = TBD).

Address:

This gives the hardware address/channel to which the Pulse Command is to be routed.

5.2.2.1 Distributing Register Load Commands

The request for the distribution of Register Load Command(s) by means of a Telecommand packet is:

Distribute Register Load Commands (2,2)

Telecommand Packet, Application Data:

N	Register Address	Register Data
Unsigned Integer	**Unsigned Integer**	**Any**
Optional, TBC	<------------------ Repeated N times ----------------->	

N:

The number of register load commands which follow (N > 0, N_max = TBD).

Register Address:

This gives the hardware address of the register.

Register Data:

The register data consist of a set of parameters whose structure is known implicitly from the foregoing register address.

5.2.2.2 Distributing CPDU commands

This Telecommand serves for executing high-priority commands directly from the TC-Decoder via its Command Pulse Distribution Unit. The request for the generation of Command Pulses on the output lines of a CPDU is:

Distribute CPDU Commands (2,3)

CPDU Telecommand Packet, Application Data:

Output Line ID	Duration		Output Line ID	Duration
Enumerated (1 octet)	**Unsigned Integer (1 octet)**		**Enumerated (1 octet)**	**Unsigned Integer (1 octet)**
1st Command Pulse Instruction			Nth Command Pulse Instruction	

Output Line ID:

This identifies the CPDU output line on which the Command Pulse is issued.

Duration:

A value between 0 and 8 which determines the duration of the Command Pulse as follows:

Command Pulse duration = <CPDU_DURATION_UNIT> * 2^{Duration}

where <CPDU_DURATION_UNIT> is defined for the CPDU (between 10 and 15 ms, TBD).

The number of Command Pulse Instructions in the CPDU Telecommand packet is variable, TBC, N_max = TBD.

5.2.3 Telemetry Packet Data Field Structure:

not applicable

5.3 Service Type 3: Housekeeping & Diagnostic Data Reporting

Periodic Housekeeping TM Packets (HK Packets) shall be used to report the status and health of each unit or subsystem on-board. In nominal conditions only **one HK Packet per unit** / subsystem / instrument should be generated with a fixed sampling period. Supercommutation within a sampling period may be used for certain parameters; they are recorded at the end of a given packet structure as a sequence of fixed-length arrays of parameters.

If a number of user-parameters need to be reported with a sampling interval significantly longer or shorter than the nominal period, or if certain HK parameters are only relevant in **specific operational modes or configurations** of a unit / subsystem / instrument, additional Housekeeping TM Packets may be implemented by allocating adequate Structure Identifiers together with Data Field definitions. The Structure Identifier defines implicitly all sampling/timing relationships for the associated packet, as well as the structure and nature of all parameters.

The generation and transmission of these nominal HK Packets shall start or stop automatically together with the corresponding mode or configuration change of a unit / application. No additional use of Service 14 (Packet Transmission Control) shall be necessary for this purpose, nor shall it be necessary to re-define Housekeeping Packets under foreseeable mission conditions.

Diagnostic TM Packets are equivalent to HK Packets in purpose, structure, and sampling approach. They may be utilised for data logging during specific modes or configurations related to calibration, engineering, or diagnostic phases (of limited duration). For easy extraction of these data on ground a special packet type is available for these data.

If certain diagnostic (or housekeeping) data need to be sampled only once (or over a short period of time) the default generation / transmission status for the involved Housekeeping / Diagnostic Packets can be “disabled”, and Service 14 may be used to activate / deactivate these packets.

5.3.1 Telecommand Packet Data Field Structure:

not applicable

5.3.2 Telemetry Packet Data Field Structure:

5.3.2.1 Reporting housekeeping or diagnostic data

The periodic reports of the values of a set of housekeeping or diagnostic parameters are:

Housekeeping Parameter Report (3,25)

Telemetry Source Packet, Source Data:

Diagnostic Parameter Report (3,26)

Telemetry Source Packet, Source Data:

SID	Parameters
Enumerated	Any

SID:

The SID, together with the Application Process ID and the nature of the packet (packet type / subtype: housekeeping or diagnostic) implicitly identifies the structure of the parameter field.

.

Parameters:

This field consists of a sequence of values of housekeeping or diagnostic parameters that are sampled nominally once per collection interval. It may be followed by a sequence of fixed-length arrays of parameters -values.

The only authorised parameter types are those described in Appendix 6.

5.4 Service Type 4: not used

5.5 Service Type 5: Event Reporting

5.5.1 Telecommand Packet Data Field Structure:

not applicable

5.5.2 Telemetry Packet Data Field Structure:

Three different sub-types of user-initiated event reports are defined for FIRST / Planck, to facilitate routing, on-board processing, and/or ground processing. All reports have the same structure, as follows:

Event Report (5,1):

This sub-type shall be used for passing on information for any **asynchronous nominal event**, that has occurred within a unit or subsystem. No direct re-action by other units, except recording or transmission, is normally required. The CDMU may decide after reception of a specific Event Packet to initiate a related nominal activity (e.g. releasing a specific Telecommand to other units).

Exception Report (5,2)

An Exception Report shall be generated by a unit in **non-nominal** cases for which an unscheduled **on-board** (recovery) **action** is required. This Report Packet is related to situations, which cannot be resolved by the unit alone but for which on-board procedures are available.

Error/Alarm Report (5,4)

An Error/Alarm Report shall be generated for **non-nominal** events which require intervention from the mission control centre **on ground**, i.e. no predefined recovery or saving procedures are resident on-board.

Data Field Structure for TM-Packet type (5,1), (5,2), and (5,4):

SID	Parameters
Enumerated 2 octets	Any
	<-------- Optional ------->

SID:

The Structure ID (SID), together with the Application Process ID and TM-Packet type, implicitly defines the nature of the event, which is reported, and the presence, structure and interpretation of the associated parameter field. Events of different subtype must have different Structure IDs.

Parameters:

The parameter field provides complementary information relating to the particular value of the Structure ID. Details of parameters and the SID are to be defined in the User Manual of the unit or subsystem.

In all cases the parameter field shall have a length corresponding to one or several 16-bit words. **The maximum length for the parameter field shall not exceed 42 octets, for a total length of the TC packet of less than 64 octets.**

5.6 Service Type 6: Memory Management

The Memory Management service covers the TC- and TM-structures needed for loading, dumping, and checking areas of on-board memories in cases where the actual contents of a memory location is not accessible otherwise. However, the basic checking of the integrity of any user memory or of the correctness of code should be part of the selftest or Error Detection And Correction (EDAC) capabilities of each on-board unit.

This service relies on the capability of a certain on-board processor to execute at least several basic tasks like TM / TC communication via the data interface, and execution of this service, in a correct way.

Different on-board processors may have different addressing capabilities, for example, some may not be capable of addressing a single octet, but instead may have an addressing granularity which corresponds to a 16-, 24- or 32-bit word. The base for the memory management service is thus referred to a Smallest Addressable Unit (SAU) , whose actual value is implementation-dependent and has to be defined in the User Manual of a subsystem / instrument.

5.6.1 Telecommand Packet Data Field Structure:

5.6.1.1 Loading data in memory using base plus offsets: not used

5.6.1.2 Loading data in memory using absolute addresses

When the user receives this Telecommand, it calculates the checksum of the received data, writes the data block to the memory at the specified start address and, if requested, re-reads the memory area just written to, calculates, compares and reports the checksum.

The request to load data to one area of a memory block defined using absolute addresses is:

Load Memory Using Absolute Addresses (6,2)

Telecommand Packet, Application Data:

Memory ID	Start Address	Length	Data	Checksum
Fixed OctetString (16 bits)	**Unsigned Integer (16 bits)**	**Unsigned Integer (16 bits)**	**Variable OctetString (N * 16 bits)**	**Fixed Bitsring (16 bits)**

Memory ID:

This identifies the destination memory block.

Start Address:

This gives the start address (in SAUs, with the count starting from zero) within the memory block for loading the data.

Length:

The number of SAUs to be loaded

Data:

A data block to be loaded (in increasing order of SAU).

Checksum:

An CRC checksum (see Appendix A.4) that is used by the on-board user to verify the integrity of the data being loaded. This checksum is generated over the entire data block of the packet, and is additional to the CRC word at the end of each packet

5.6.1.3 Dumping memory using base plus offsets: not used

5.6.1.4 Dumping memory using absolute addresses

When an on-board user receives this request it reads the memory block, generates a report packet containing the contents of this area and downlinks it.

The request to dump the contents of one area of a memory block defined using absolute addresses is:

Dump Memory Using Absolute Addresses (6,5)

Telecommand Packet, Application Data:

Memory ID	Start Address	Length
Fixed OctetString (16 bits)	**Unsigned Integer (16 bits)**	**Unsigned Integer (16 bits)**

Memory ID:

This identifies the destination memory block of the on-board user.

Start Address:

This gives the start address (in SAUs, with the count starting from zero) within the memory block for dumping the data.

Length:

The number of SAUs to be loaded

5.6.1.5 Checking memory using absolute addresses

When the Service Provider receives this request it reads and computes the checksum value of the indicated area of the memory using the CRC checksum algorithm defined in Appendix A.4. It then generates a report containing the checksum value computed.

The request to check the contents of one area of a memory block defined with absolute addresses is:

Check Memory Using Absolute Addresses (6,9)

Telecommand Packet, Application Data:

Memory ID	Start Address	Length
Fixed OctetString (16 bits)	Unsigned Integer (16 bits)	Unsigned Integer (16 bits)

Memory ID, Start Address, Length: equivalent to para. 5.6.1.4

5.6.2 Telemetry Packet Data Field Structure:

5.6.2.1 Memory Dump, Absolute Addresses (6,6)

Telemetry Source Packet, Source Data:

Memory ID	Start Address	Length	Data	Checksum
Fixed OctetString (16 bits)	Unsigned Integer (16 bits)	Unsigned Integer (16 bits)	Variable OctetString (16 bits)	Fixed Bitsring (16 bits)

Memory ID, Start Address, Length: as for para. 5.6.1.4

Data:

The data block to be dumped (in increasing order of SAU).

Checksum:

The Service calculates an CRC checksum according to Appendix A.4 for the data being dumped and places the result in this field.

5.6.2.2 Memory Check Report,Absolute Addresses (6,10)

Telemetry Source Packet, Source Data:

Memory ID	Start Address	Length	Checksum
Fixed OctetString (16 bits)	Unsigned Integer (16 bits)	Unsigned Integer (16 bits)	Fixed BitString (16 bits)

Memory ID, Start Address, Length: equivalent to para. 5.6.1.4

Checksum:

The Service calculates an CRC checksum according to Appendix A.4 for the data being checked and places the result in this field.

5.7 Service Type 7: Task Management

This section defines the application data structures for Task Management Telecommand and Telemetry Packets.

According to the Packet Utilisation Standard, **[RD 1]**, the term **“Task”** refers to software processes, which execute at a lower level than an Application Process. These software processes typically execute under the control of an operating system or a run-time kernel, on behalf of a given Application Process, and several tasks may be active in parallel or may together constitute a Function.

The ground will **normally not directly interact with these tasks**, however for contingencies or troubleshooting purposes, the ground may have the capability to exercise direct control over **some** of these processes. The term “task” is used to denote these implementation-specific low-level processes, in contrast to **Functions** (Service 8) or **On-board Control Procedures** (Service 18).

It may be that a certain unit or application can fulfil all required functionality by interfacing with other applications or the ground by utilising Service 8 (and 18) only. In this sense the Service 7 is **optional.**

5.7.1 Telecommand Packet Data Field Structure:

5.7.1.1 Starting a task

When the request is received without following parameters, the on-board application shall start the specified task with default parameters. If a number of parameters is present in the Telecommand, these parameters shall be used instead.

In all cases the task status shall become “running”. The request is ignored if the task status was "running" or “suspended”.

The request is:

Start Task (7,1)

Telecommand Packet, Application Data:

Task ID	N	Parameters
8 bits	Unsigned Integer 8 bits	Any
		Optional

Task ID (TID):

The Task ID, together with the Application ID in the packet header, implicitly defines the presence and the fixed structure of the Parameter Field which follows.

The Task ID is to be defined in the User Manual of the subsystem or instrument.

N:

This parameter specifies the number of the parameters which follow.

Parameters:

A number of data structures compliant with the Structure Rules according to Appendix 6. The parameters are used to configure the specific instance of the execution of the task. The structure of the Parameter Field is implicitly defined by the Task ID and Application ID, and it contains a maximum of N_max parameters.

If N is smaller than N_max then the first N parameters of the Parameter Field shall be sent to the task, the other parameters shall retain their current/last value.

5.7.1.2 Stopping a task

The request is:

Stop Task (7,2)

Telecommand Packet, Application Data:

Task ID	Spare
8 bits	8 bits

When this request is received, the on-board Application Process stops the specified task. The task status is then "stopped". The request is ignored if the task status was "stopped".

5.7.1.3 Suspending a task

If the actual status of a Task is known it may be suspended (during testing phases).

The request is:

Suspend Task (7,3)

Telecommand Packet, Application Data:

The same as for the "Stop a Task" Request.

When this request is received, the on-board Application Process suspends the specified task. The task status is then "suspended" and the task context is frozen.

The request is ignored if the status of the task was "stopped" or "suspended".

5.7.1.4 Resuming a task

The request is:

Resume Task (7,4)

Telecommand Packet, Application Data:

The same as for the "Stop a Task" Request.

When this request is received, the on-board Application Process resumes the specified task from the point where it was previously suspended. The task status is then "running. The request is ignored if the status of the task was "stopped" or "running".

5.7.1.5 Aborting a task

Any on-board task shall be designed such that it can **always be terminated in a nominal way**, which leaves the task and related higher-layer processes in a predictable state. Therefore, the Stop Task Telecommand (7,2) should always be sufficient for ending a task.

As it cannot be verified that in any non-nominal case the execution of an abort command will leave the involved unit or subsystem in a state, from which it can continue to operate predictable or safely during a space-mission, the abort task command shall not be used.

5.7.2 Telemetry Packet Data Field Structure:

not applicable

5.8 Service Type 8: Function Management

This section defines the application data structures for Function Management Telecommand and Telemetry Packets.

According to the Packet Utilisation Standard, **[RD 1]**, the term **"Function"** refers to software processes, which execute at a higher level than Tasks. The ground will normally have the capability to exercise direct control over these processes.

An Application Process may be designed to execute Functions in support of (external,) physical interfaces and to control the spacecraft-related functions of these interfaces. (The actual I/O-handler may be a Task as part of the Function.) Examples of such Application Functions could include control of the operation of a payload instrument or spacecraft units and subsystems, including changes of operational modes. Additionally, processes that are (mainly) devoted to internal data processing, and which rely on nominal control capability from ground, can be covered under the Function Service.

When an application function is active, the ground may have the possibility to perform a number of activities in the current context of the function. For example, in a particular mode of operation of an instrument, certain specific control actions may be available.

An important characteristic of the Function Management Service is, that it can be implemented in a way that a **two-step "Arm – Fire-"** or "Enable – Execute-"concept can be utilised, additionally to the immediate execution after reception of a Start Function Telecommand (8,1). This is achieved by separating the loading of new function-parameters from starting a Function with these parameters. Also, for establishing precise control over the timing of Functions or Activities these Telecommands can be used.

5.8.1 Telecommand Packet Data Field Structure:

5.8.1.1 Activating a Function

When the request is received without following parameters, the on-board application shall start the specified Function with default parameters. If a number of parameters is present in the Telecommand, these parameters shall be used instead.

In case a Load Function Parameters command TC(8,8) has been received prior to the Start Telecommand, the Function shall be activated with these parameters.

In all cases the Function status shall become "running". The request is ignored if the Function status was "running".

The request is:

Start Function (8,1)

Telecommand Packet, Application Data:

Function ID	N	Parameters
Enumerated 8 bits	Unsigned Integer 8 bits	Any
		Optional

Function ID (FID):

The Function ID, together with the Application ID in the packet header, implicitly defines the presence and the fixed structure of the Parameter Field which follows.

The Function ID is to be defined in the User Manual of the subsystem or instrument.

N:

This parameter specifies the number of the parameters which follow.

Parameters:

A number of data structures compliant with the Structure Rules according to Appendix 6. The parameters are used to configure the specific instance of the execution of the Function. The structure of the Parameter Field is implicitly defined by the Function ID and Application ID, and it contains a maximum of N_max parameters.

If N is smaller than N_max then the first N parameters of the Parameter Field shall be sent to the Function, the other parameters shall retain their current/last value.

5.8.1.2 Stopping a Function

The request is:

Stop Function (8,2)

Telecommand Packet, Application Data:

Function ID	Spare
Enumerated 8 bits	8 bits

When this request is received, the on-board Application Process stops the specified Function. The Function status is then "stopped". The request is ignored if the Function status was "stopped".

5.8.1.3 Performing an activity of a function

When this request is received, the Application Process indicates to the Function which activity it must perform and the parameters to be used. The Function status is unchanged by the execution of the Activity. If a certain Function does not need to be split into more than one Activities, TC type (8,3) can be ommitted.

If the status of a Function is **“running”** it shall be possible to load parameters by executing TC(8,3), and have them accepted immediately without changing the Function status.

The request is ignored if the Function has the "stopped" status.

The request is:

Perform Activity of Function (8,3)

Telecommand Packet, Application Data:

Function ID	Activity ID	N	Parameter
Enumerated	Enumerated	Unsigned Integer	Any
			<----Repeated N times ---->

Function ID (FID):

The Function ID, together with the Activity ID and the Application ID in the packet header, implicitly defines the presence and the fixed structure of the Parameter Field which follows.

Activity ID:

This indicates which activity of the specified Application Function is to be performed.

N:

The number of Parameters which follow.

Parameter:

A number of data structures compliant with the Structure Rules according to Appendix 6. The parameters are used to configure the specific instance of the execution of the Function. The structure of the Parameter Field is implicitly defined by the Function ID and Application ID, and it contains amaximum of N_max parameters.

If N is smaller than N_max then the first N parameters of the Parameter Field shall be sent to the Function, the other parameters shall retain their current/last value.

5.8.1.4 TC (8,4) : Load Function Parameters

When the request is received, the specified functional parameters shall be made available to the Function in order to perform an activity within a Function.

As it may be essential to do a **verification** before execution or to have **exact timing control** over a Function, the Load Function Parameters Telecommand can be executed. If the status of the Function is **“stopped”,** the new parameter values loaded by a Load Function Parameters TC(8,4) command shall be stored. In a second step, the reception of a Activate Function TC(8.1) Telecommand, these parameters shall replace any old or default parameter values and the status of the Function shall change to “running”.

The request is:

Load Task Prameters (8,4)
Telecommand Packet, Application Data:

Function ID	Activity ID	N	Parameter
Fixed CharString	Enumerated	Unsigned Integer	Any
			<----Repeated N times ---->

Function ID (FID):

The Function ID, together with the Activity ID and the Application ID in the packet header, implicitly defines the presence and the fixed structure of the Parameter Field which follows.

Activity ID:

This indicates which activity of the specified application function is to be performed. Thus the Activity ID, together with the Function ID and the Application Process ID in the packet header, implicitly defines the presence of, and the structure of, the Parameter Field which follows.

N:

This parameter specifies the number of the parameters which follow.

Parameters:

A number of data structures compliant with the Structure Rules according to Appendix 6. The parameters are used to configure the specific instance of the execution of the Function. The structure of the Parameter Field is implicitly defined by the Function ID and Application ID, and it contains a maximum of N_max parameters.

If N is smaller than N_max then the first N parameters of the Parameter Field shall be sent to the Function, the other parameters shall retain their current/last value.

5.8.2 Telemetry Packet Data Field Structure:

not applicable

5.9 Service Type 9: Time Management

The Time Management Service provides the capabilities to transfer a master time information on-board a spacecraft, the Central Time Reference (CTR), to other units or applications after activation, reset, or for the purpose of re-synchronisation in order to achieve a high accuracy of “local copies” of the CTR. Additionally, the service supports the verification of time information, which is maintained (independently) within applications, against the on-board master time.

This is accomplished by executing a time synchronisation procedure in the CDMS, and locally within the addressed end-user or Application. This consists of sending an Enable Time Synchronisation packet TC(9,4) and a Time Code TC(9,5) to this Application, and then sending a synchronisation signal or message to one or more end-users. The submitted Time Code is valid at this moment. Also lower communication layers may be involved in time-setting and synchronisation of end-users, see also Appendix 9 of this document.

A similar process may be used to (re-)set the Central Time Reference on-board the spacecraft to a ground reference time.

Neither the operation of a (local) on-board clock nor the time-tagging of TM packets shall be suspended at any time.

The verification of the on-board master time against a time reference on-ground is according to **[AD 1]** a function, which is carried out by the CDMS on Transfer Frame Layer, and involves the generation of Standard Spacecraft Time Source Packets from the reserved Application ID Zero. This procedure is not covered in this section, for details see Appendix 7.

5.9.1 Telecommand Packet Data Field Structure:

5.9.1.1 TC (9,3) : Synchronise User

On receipt of this command, the CDMS shall execute the time synchronisation procedure.

Application Data :

Spare	Application ID
5 bits	11 bits
mandatory	mandatory

Spare :

Set to all zeros, i.e. '00000'$_{\mathrm{BIN}}$.

Application ID:

This field carries the APID of the user to be synchronised.

5.9.1.2 TC (9,4) : Enable Time Synchronisation

The CDMS sends this command to the user addressed, to start the local time synchronisation procedure. The user then waits for a Time Code Telecommand and the next synchronisation signal or message and then sets its local time to the time delivered in the Time Code Telecommand.

Application Data : None

5.9.1.3 TC (9,5): Time Code

The user sets its local clock to the time information delivered in the data field, in order to generate a copy of the on-board master time. The exact moment in time, at which this information is valid, is determined by the next synchronisation signal on the on-board data bus. User-internal latencies or corrections may be taken into account within the process of local time synchronisation, depending on the required accuracy.

Application Data :

Time
48 bits
mandatory

Time : (48 bits)

The copy of the CDMS master time (CTR) to be added. The time code format shall be as specified in Appendix 6.3.6.

5.9.1.4 TC (9,6) : Verify User Time

On receipt of this command, the CDMS shall generate an Enable Time Verification packet TC(9,7) for the user identified in the data field and then execute the time verification procedure, which includes the generation of a Central time Reference packet TM (9,8).

Application Data :

Spare	Application ID
5 bits	11 bits
mandatory	mandatory

Spare :

Set to all zeros, i.e. '00000'$_{BIN}$.

Application ID:

This field carries the APID of the user to be verified.

5.9.1.5 TC (9,7) : Enable Time Verification

On receipt of this command, the application addressed shall enable the generation of a Time Verification Report TM(9,9)

Application Data : None

5.9.2 Telemetry Packet Data Field Structure: Onboard Time Management Reports

The Time Management Reports are to be used by the CDMS and selected onboard applications to verify synchronisation to a common onboard time reference.

5.9.2.1 TM (9,8) : Central Time Reference

Within the time verification procedure the CDMS shall generate a Central Time Reference packet TM(9,8) with a copy of the CDMS Central Time Reference generated at the moment of the next synchronisation signal, which is sent to end-users.

Application Data :

Time
48 bits
mandatory

Time : (48 bits)
The copy of the Central Time Reference managed by the CDMS. The time code format shall be as specified in Appendix 6.3.6.

5.9.2.2 TM (9,9) : Time Verification Report

After reception of a TC(9,7) the application addressed shall generate a Time Verification Report TM(9,9) at the moment of reception of the next synchronisation signal. The data field shall carry a copy of the local user time.

Application Data :

Time
48 bits
mandatory

Time : (48 bits)
The copy of the Central Time Reference managed locally by the application. The time code format shall be as specified in Appendix 6.3.6.

5.10 Service Type 10: not used

5.11 Service Type 11: On-board Scheduling

The On-board Scheduling Service provides for the capabilities to execute a sequence of time-tagged Telecommands from a Mission Timeline (MTL). These Telecommands are passed by the CDMU to end-users, identified by their APIDs, at execution time, provided that the schedule interlock conditions for a certain Telecommand are met.

If an error is detected during the releasing of a Telecommand for execution, it shall not affect the processing of the remainder of the Telecommands of the MTL. The release of these Telecommands shall depend on the preconditions for these Telecommands, determined at execution time.

Additionally the Service 11 provides for means to insert, delete, and time-shift Telecommands in the MTL, and to report the contents of the MTL. Details of the service concept are outlined in the PUS, **[RD 3].**

If the insertion, deletion, or time-shifting of a MTL-Telecommand results in an error, e.g. logical conflict with the rest of the schedule, this operation shall not be carried out.

5.11.1 Telecommand Packet Data Field Structure:

5.11.1.1 Controlling the release of Telecommands

The Service Requests to enable or disable the release of selected Telecommands are:

Enable Release of Telecommands (11,1)

Telecommand Packet, Application Data:

Disable Release of Telecommands (11,2)

Telecommand Packet, Application Data:

N	Application Process ID
Unsigned Integer	**Enumerated**
	<-------- Repeated N2 times ------->

N:

The number of Application IDs that follow.

If **N = 0**: the execution of **the MTL for all units/ applications** on-board is enabled / disabled.

Application Process ID:

The identification of the destination Application Process(es) for which the MTL is to be enabled/disabled.

When the CDMS receives this request, then the schedule level controlling attribute of all Telecommands is set according to the request type. If it is an enable request, then the Command Schedule Scheduling Event time is also set.

Execution of the MTL shall start from the actual value of the master on-board time (CTR) onwards.

5.11.1.2 Resetting the Command Schedule

The request is:

Reset Command Schedule (11,3)

Telecommand Packet, Application Data: **None**

When the Service Provider receives this request:

- it clears all entries in the Command Schedule (MTL) for all Application Processes;
- it resets the interlock information (no interlock defined);
- it resets the Scheduling Event information (all Scheduling Event times are unknown).

5.11.1.3 Inserting MTL-Telecommands in the Command Schedule

With this command one Telecommand for a certain end-user, identified by its APID, can be added to the on-board Mission Timeline, together with the conditions for release, and the time of release. This Telecommand type serves for **inserting** specific MTL-Telecommands into the Command Schedule, **and for loading the complete MTL** for a longer period of operation, by sending all MLT-Telecommands for that period.

With each TC(11,4) one executable Telecommand shall be loaded. The actual size of executable Telecommands is limited to TBD octets (TBD octets for the data field) because the data space available to end-users is reduced by the schedule control parameters.

Note: As FIRST and Planck will be operated from the MTL most of their time this restriction applies for the majority of all Telecommands.

The request to insert (e.g. add) one Telecommand in the Command Schedule is:

Insert MTL-Telecommands in Command Schedule (11,4)

Telecommand Packet, Application Data:

Interlock Set ID	Interlock Assessed ID	Assessment Type	Scheduling Event	Abs/Rel Time Tag	Execution Timeout	Telecommand Packet
Enumerated	Enumerated	Enumerated	Enumerated	Absolute or Relative Time	Relative Time	Variable OctetString
Optional		Optional	Optional		Optional	

Interlock Set ID:

The identification of the interlock to be set by this Telecommand (0 if no interlock is to be set). The status of this interlock will be determined by the success or failure of the Telecommand execution.

This field is systematically omitted if the Service does not support the concept of interlocking.

Interlock Assessed ID:

The identification of the interlock on which the release of this Telecommand is dependent (0 if no interlock is to be assessed).

This field is systematically omitted if the Service does not support the concept of interlocking.

Assessment Type:

This determines whether the release of this Telecommand is dependent on the success or failure of the Telecommand with which it is interlocked:

"Success" (value = 1): release if interlocking Telecommand was successful;

"Failure" (value = 0): release if interlocking Telecommand failed execution.

Scheduling Event:

This determines whether the release time of this Telecommand is an absolute on-board CUC/TAI time (Scheduling Event = "Absolute", value = 0) or a relative time and, in the latter case, this parameter indicates the type of Scheduling Event for the relative time.

If the Scheduling Event = "Schedule" (value = 1), the Telecommand release time is relative to the time at which the Command Schedule is enabled.

If the Scheduling Event = "Interlock" (value = 3), then the Telecommand release time is relative to the time of notification to the Service of the success or failure of the Telecommand which sets the interlock.

This field is systematically omitted if the Service does not support the concept of relative time.

Abs/Rel Time Tag:

If Scheduling Event = "Absolute", then this is the on-board CUC/TAI time at which the Telecommand packet is to be sent to its Application Process ID. The format and length of this field are TBD.

If the Scheduling Event indicates a relative time, then this is an on-board positive delta time which, when added to the time of occurrence of the event identified by the Scheduling Event, determines the absolute time at which the Telecommand packet is to be sent to its destination Application Process. The format and length are the same as for the absolute time.

Execution Timeout:

This is an on-board positive delta time which, when added to the time of release of the Telecommand, determines the latest time at which the Telecommand is expected to complete execution. This parameter is only present if the Telecommand sets an interlock.

If an execution completion report is not received by the On- board Scheduling Service within the timeout window, then the Telecommand is deemed to have failed. Also, if the Telecommand defines a Scheduling Event, then the Scheduling Event time is set to the timeout window upper bound and the absolute times of the related relative time Telecommands become known (they would otherwise remain indefinitely in the Command Schedule).

The format and length of this field are the same as for the absolute time tag.

Telecommand Packet:

This is a standard Telecommand packet of any Type/ Subtype

The source of the Telecommand packet is indicated in the Source Part of the Packet Sequence Control field, its value is 001 (i.e. Telecommand from MTL) .

If a Telecommand to be added has a relative time with respect to an interlock, it is "linked" to the latest Telecommand added to the Command Schedule which sets that interlock.

When this request is received, the Telecommand in the request is checked for consistency with the rest of the MTL and, if no error is detected, it is added to the Command Schedule.

5.11.1.4 Deleting Telecommands from the Command Schedule

The On-board Scheduling Service will refuse to delete an interlocking Telecommand unless all its (directly and indirectly) interlocked Telecommands have either already been deleted or are deleted in the same deletion request.

If an error is detected during the processing of a deletion request, no Telecommands are deleted.

5.11.1.4.1 Deleting a Set of Telecommands from the MTL

The request to delete sets of Telecommands from the Command Schedule is:

Delete MTL-Telecommands (11,5)

Telecommand Packet, Application Data:

N	Application Process ID	Sequence Count	Number of Telecommands
Unsigned Integer	Enumerated	Enumerated	Unsigned Integer
Optional	<---------------------------- Repeated N times ---------------------------->		

N:

This field specifies the number of Applications for which a sequence of Telecommands shall be deleted.

Application ID:

The APID identifies the Application affected.

Sequence Count:

The identification of the first Telecommand packet to be sent to the specified destination Application Process, which is to be deleted.

The doublet (Application Process ID, Sequence Count) uniquely identifies a Telecommand packet.

Number of Telecommands:

The number of successive Telecommand packets sent by the CDMU to the specified destination Application Process which are to be deleted from the MTL.

When this request is received, all Telecommands which satisfy the selection criteria defined by the Application Process ID, Sequence Count and the Number of Telecommands are deleted.

The deletion of Telecommands which have times relative to Scheduling Events, which have not yet occurred, can only be performed by means of this type of request (11,5).

5.11.1.4.1 Deleting Telecommands over a time period

The request is:

Delete Telecommands over Time Period (11,6)

Telecommand Packet, Application Data:

Range	Time Tag 1	Time Tag 2	N2	Application Process ID
Enumerated	Absolute Time	Absolute Time	Unsigned Integer	Enumerated

<-- Repeated N2 times -->

Range:

This indicates whether the time period is :

- from the beginning to the end of the Command Schedule if Range is ***"All"*** (**value = 0**) or
- between Time Tag 1 and Time Tag 2 inclusive if Range is ***"Between"*** (**value = 1**)

Time Tag 1:

The earliest absolute time if Range is ***"Between"*** .

Time Tag 2:

The latest absolute time if Range is ***"Between"****.*

N2:

This field specifies the number of Applications for which a sequence of Telecommands shall be deleted.

Application Process ID:

The identification of the destination Application Process from which Telecommands are to be deleted.

Those Telecommands whose absolute release times are not yet known are not deleted from the Command Schedule. A Telecommand has an unknown release time if it has a time relative to (directly or indirectly) a Scheduling Event which has not yet occurred.

5.11.1.5 Time-shifting of Telecommands in the command schedule

The time-shift request contains the time offset to be added (which may be a positive or negative value), and specifies the group of Telecommands to which this time-offset is to be applied by specifying the start and end time of the time interval for which MTL Telecommands have to be shifted.

The Scheduling Service will refuse to time-shift a Telecommand if its new absolute time would fall in the past or before the end of the execution window of its interlocking Telecommand (if it is interlock dependent) or if its new relative time would become negative.

5.11.1.5.1 Time-shifting Telecommands

The request to time-shift sets of Telecommands in the Command Schedule is:

Time-Shift Telecommands (11,7)

Telecommand Packet, Application Data:

Time Offset	N	Application Process ID	Sequence Count	Number of Telecommands
Relative Time	Unsigned Integer	Enumerated	Enumerated	Unsigned Integer
		<---------------------- Repeated N times ------------------------>		

Time Offset:

A positive or negative interval of time expressed in the length and format of relative time defined for the Service or mission (since it is the relative time between the new and the old values of release time).

N:

This field specifies the number of Applications for which a sequence of Telecommands shall be shifted.

When this request is received the release times in the Command Schedule are modified (by adding the specified time offset) for those Telecommands which meet the selection criteria defined by the specified Application Process ID, Sequence Count and Number of Telecommands. An error occurs if the first Telecommand to be time-shifted is not found in the Command Schedule.

In the case of a Telecommand with relative release time, it is the relative time which is modified if its Scheduling Event has not yet occurred.

5.11.1.5.2 Time-shifting Telecommands over a time period

The request is:

Time-Shift Telecommands over Time Period (11,8)

Telecommand Packet, Application Data:

Range	Time Tag 1	Time Tag 2	Time Offset
Enumerated	Absolute Time	Absolute Time	Relative Time
	Optional	Optional	

N	Application Process ID

Unsigned Integer	**Enumerated**
	Repeated N2 times

Range:

The value of Range is 1.

N:

This field specifies the number of Applications for which a sequence of Telecommands shall be shifted.

When this request is received, the release times of the following Telecommands are time-shifted if they have release times falling in the specified absolute time period.

Those Telecommands whose absolute release times are not yet known are not time-shifted. A Telecommand has an unknown release time if it has a time relative to (directly or indirectly) a Scheduling Event which has not yet occurred.

5.11.1.6 Reporting of the Command Schedule contents

This subservice provides the capabilities to produce a summary or detailed report of the status and contents of the MTL.

Detailed reporting of the command schedule

The request to obtain a detailed report on sets of Telecommands in the Command Schedule is:

Report Command Schedule in Detailed Form (11,9)

Telecommand Packet, Application Data:

N	Application Process ID	Sequence Count	Number of Telecommands
Unsigned Integer	**Enumerated**	**Enumerated**	**Unsigned Integer**
Optional	<----------------------- Repeated N times ------------------------>		

N, APID, Sequence Count, Number of Telecommands:

The structure and function of these fields is equivalent to type/subtype (11,5).

When this request is received, a Report of type/subtype (11,10) is generated containing those Telecommands in the Command Schedule which meet the selection criteria defined by the combination of Application Process ID, Sequence Count and Number of Telecommands.

Summary reporting of the Command Schedule

The request to obtain a summary report of sets of Telecommands in the Command Schedule is:

Report Command Schedule in Summary Form (11,12)

Telecommand Packet, Application Data:

Same as for the "Report Command Schedule in Detailed Form" Telecommand, type (11,9).

When this request is received, a Report of type (11,13) is generated containing those Telecommands in the Command Schedule which meet the selection criteria defined by the combination of Application Process ID, Sequence Count and Number of Telecommands

Summary reporting of the MTL over a time period

The request for a summary report of selected part(s) of the Command Schedule over an absolute time period is:

Report Command Schedule in Detailed Form over Time Period (11,14)

Telecommand Packet, Application Data:

Range	Time Tag 1	Time Tag 2	N2	Application Process ID
Enumerated	Absolute Time	Absolute Time	Unsigned Integer	Enumerated
			Optional	Repeated N2 times

Parameters of Data Field:
The structure and function of these fields is equivalent to type/subtype (11,6.).

- When this request is received, a **Summary Schedule Report** is generated.

Telecommands whose absolute release times are not yet known are not included in the report. A Telecommand has an unknown release time if it has a time relative to (directly or indirectly) a Scheduling Event which has not yet occurred.

5.11.2 Telemetry Packet Data Field Structure:

5.11.2.1 Detailed Report of the Command Schedule

The Report contains all the static scheduling attributes for the selected Telecommands. Telecommands whose release time is not yet known can only be selectively reported by means of this type of report.

Detailed Schedule Report (11,10)

Telemetry Packet, Source Data:

Interlock Set ID	Interlock Assessed ID	Assessment Type	Scheduling Event	Abs/Rel Time Tag	Execution Timeout	Telecommand Packet
Enumerated	Enumerated	Enumerated	Enumerated	Absolute or Relative Time	Relative Time	Variable OctetString
Optional		Optional	Optional		Optional	

Parameters of Data Field:

The structure and function of these fields is equivalent to type/subtype (11,4), i.e. for the selected MTL-Telecommands the contents of the MTL is reported the same way as it was loaded or inserted.

5.11.2.2 Summary Report of the Command Schedule

The Report contains only the identifications for the selected Telecommands. Telecommands whose release time is not yet known can only be selectively reported by means of this type of request.

Summary Schedule Report (11,13)

Telemetry Source Packet, Source Data:

N	Scheduling Event	Abs/Rel Time Tag	Application Process ID	Sequence Count
Unsigned Integer	Enumerated	Absolute or Relative Time	Enumerated	Enumerated
	Optional	Optional		
	<--- Repeated N times --->			

Parameters of Data Field:

The structure and function of these fields is equivalent to type/subtype (11,10.).

5.11.2.3 Summary Report of the Command Schedule over a time period

Telemetry Packet, Application Data:

Identical to the "Summary Schedule Report" , TM packet type (11,13).

5.12 Service Type 12: On-board Monitoring

The On-board Monitoring Service provides the **capability to monitor on-board parameters** with respect to checks defined by the ground **for the purpose of initiating adequate actions** on-board (during autonomy phases of the spacecraft) or on ground. These actions can be confined to reporting in some cases, or predefined on-board functions (like nominal OBCPs, or FDIR functions) may be involved, if applicable. In any case the service reports **all** check status transitions to the ground.

To achieve this, the service maintains a monitoring list, and checks parameter samples according to the information contained therein. If a check results in a positive result, an Event packet (type 5,x) shall be generated, which can be evaluated on-board by the Event/ Action Service (Type 19), or on ground in order to initiate related further activities. The Monitoring Service is considered to be a CDMS capability, as far as S/C-related parameters are concerned. For instrument-internal monitoring this service may be implemented as part of the instrument applications.

The on-board Monitoring Service maintains static monitoring information for each parameter to be monitored. The **parameter monitoring information** specifies:

- The Identification of the on-board parameter to be monitored;
- The Monitoring Status, stating whether the monitoring of the parameter is enabled or disabled (normally **controlled by ground**);
- The parameter monitoring information includes a set of **Check Definitions**. A check definition provides the information required to checking a sample of the parameter against either one pair of limits, or one expected value. Two check definitions may be associated with a given parameter if a certain action is defined for a set of Warning limits (typically reporting), and more severe Alarm limits (typically starting a recovery action).

If different sets of check definitions for a certain parameter are associated with different operational conditions of a unit or application, the check definitions and the monitoring status shall be changed adequately and timely by commanding (from ground or CDMS/ OBCP). This includes the power status of units and activation status of (SW-) applications /functions.

In this sense the samples of a parameter, for which monitoring is enabled, are always valid, and the monitoring shall not result in a monitoring event under nominal conditions, in contrast to unscheduled or non-nominal behaviour of a parameter.

For all parameters to be monitored a **checking status list** shall be maintained, which reflects for all parameters the current/last status of the checks performed on them (i.e. “within limits”, “below low limit”, “above high limit”, etc.). If a **check status transition** occurs for a parameter, an Event TM packet (Type 5,x) shall be generated immediately, i.e. both the exceeding of limits and returning into the nominal range shall be reported (equivalently for an expected-value-check).

For a limit-check or an expected-value-check, the number of successive samples of the parameter which must fail (or succeed) the check in order to establish a new checking status for the parameter shall be fixed (nominal value: 2, TBC). Repetitive generations of identical Event packets must be avoided, details are defined in **[AD 3]**.

The monitoring intervals for these parameters shall be equal to the sampling periods for these parameters for the Housekeeping & Diagnostics Service (Service 3).

5.12.1 Telecommand Packet Data Field Structure:

5.12.1.1 Controlling the on-board monitoring

It is possible to enable or disable the monitoring of parameters globally or to enable or disable the monitoring of a specified subset of parameters. The requests are:

Enable Monitoring of Parameters (12,1)

Telecommand Packet, Application Data:

Disable Monitoring of Parameters (12,2)

Telecommand Packet, Application Data:

N	Parameter-ID
Unsigned Integer	**Enumerated**
Optional	<------- Repeated N times -------->

N:

The number of parameters whose monitoring is to be enabled/disabled. By convention, N = 0 means "Enable/disable the entire Monitoring Service ".

Parameter-ID:

The identification of a parameter.

When the CDMS receives this request:

- If **N = 0**, it sets the Service-level monitoring status to "Enabled" or "Disabled", depending on the request sub-type.
- If N > 0 and if "enable" is requested, these parameters in the Monitoring List start being monitored.
- If N > 0I and if "disable" is requested, the monitoring of these parameters in the Monitoring List is stopped. The current content of the Transition Reporting List is not affected by these activities.

An error shall be flagged if a certain parameter is not in the list. However, the processing of the remaining parameters is not affected.

5.12.1.2 Clearing the monitoring list

The request is:

Clear Monitoring List (12,4)

Telecommand Packet, Application Data: **None**

When the CDMS receives this request, it sets the Service monitoring status to "Disabled" and clears all entries in the Monitoring List and in the Transition Reporting List.

5.12.1.3 Adding Parameters to, and Modifying Parameters of the Monitoring List

The request is:

Modify Monitoring List (12,5)

Telecommand Packet, Application Data:

N	Parameter-ID	Parameter Monitoring Status
Unsigned Integer	Enumerated	

<-------------------- Repeated N times -------------------------

NOL	Low Limit	High Limit
Unsigned Integer	Deduced	Deduced

<------------ Repeated NOL times --------------->

---------------------------- Repeated N times (contd) -----------------------------

NOE	Expected Value
Unsigned Integer	Deduced

<---- Repeated NOE times ----->

------------------ Repeated N times (contd) --------------->

N:

The number of parameters to be added to the Monitoring List or to be modified.

Parameter-ID:

The identification of a parameter to be monitored.

Parameter Monitoring Status:

This indicates whether the monitoring of the corresponding parameter shall be enabled (value = 1) or disabled (value = 0).

NOL, NOE:

The number of limit-check definitions (or expected-value-check definitions) which follow. NOL can be 0, 1 or 2, NOE should be 0 or 1, TBC.

Low Limit, High Limit, Expected Value:.

The type and format of the Low Limit, High Limit, or Expected Value are the same as the type and format of the values of the parameter to be monitored.

When the CDMS receives this request, it adds the parameter monitoring information to the Monitoring List, and sets the parameter monitoring status according to the value provided in the Monitoring Status field. In case a parameter has already an entry in the Monitoring list, the values provided with the new Telecommand shall replace the previous values, in order to facilitate a modification. Consequently the parameters NOL or NOE are constant for a given parameter. In case of a monitoring parameter modification the monitoring status shall also be set to "Disabled".

If an error is detected during the processing of the monitoring information for a given parameter, this parameter is not added to the Monitoring List. This does not affect the processing of the remaining parameters

5.12.1.4 Deleting parameters from the monitoring list

The request to delete specified parameters from the monitoring list is:

Delete Parameters from Monitoring List (12,6)

Telecommand Packet, Application Data:

N	Parameter-ID
Unsigned Integer	**Enumerated**
	<-- Repeated N times -->

N:

The number of parameters to be added to the Monitoring List.

When the CDMS receives this request, it processes each parameter in turn and removes its corresponding monitoring information, if any, from the Monitoring List (the entry becomes free).

If a certain parameter is not in the Monitoring List there shall be no effect on the deletion of the parameters which have an entry in the Monitoring List.

5.12.1.5 Request for the current monitoring list contents

If there is uncertainty on ground about the current contents and setting of the on-board monitoring list the ground may request a dump of the monitoring list.

The request is:

Report Current Monitoring List (12,8)

Telecommand Packet, Application Data: **None**

When the CDMS receives this request, it issues a report with the current static contents of the Monitoring List (TM packet type (12,9)).

5.12.1.6 Request for the current parameters out-of-limit list

If the information provided by previously evaluated (Monitoring) Event packets is deemed to be insufficient the ground may request an actual summary Out-of-Limits List.

The request is:

Report Current Parameters Out-of-Limit List (12,10)

Telecommand Packet, Application Data: **None**

When the CDMS receives this request, it issues a summary-report(TM type(12,11)) for the parameters, for which the current checking status of the parameter is equal to "Below low limit", "Above high limit", or "Expected Value". Thus a parameter may appear more than once in the report (e.g. if it violates two of its check definitions).

For all parameters the first time, at which the transition of the corresponding limit check has occurred, or at which a match of the actual value with the expected value has occurred, is reported. Additionally, the actual value of each parameter is reported.

5.12.2 Telemetry Packet Data Field Structure:

5.12.2.1 Reporting the current monitoring list contents

Current Monitoring List Report (12,9)

Telemetry Source Packet, Source Data:

N	Parameter-ID	Parameter Monitoring Status
Unsigned Integer	Enumerated	

<-------------------- Repeated N times -------------------------

NOL	Low Limit	High Limit
Unsigned Integer	Deduced	Deduced

<------------ Repeated NOL times --------------->

--------------------------- Repeated N times (contd) ----------------------------

NOE	Expected Value
Unsigned Integer	Deduced

<---- Repeated NOE times ----->

----------------- Repeated N times (contd) -------------->

N:

The number of parameters of the Monitoring List. to be reported.

Parameter-ID:

The identification of a parameter to be monitored.

Parameter Monitoring Status:

This indicates whether the monitoring of the corresponding parameter is currently enabled (value = 1) or disabled (value = 0).

NOL, NOE:

The number of limit-check definitions (or expected-value-check definitions) which follow. NOL can be 0, 1 or 2, NOE should be 0 or 1, TBC.

Low Limit, High Limit, Expected Value:.

The type and format of the Low Limit, High Limit, or Expected Value are the same as the type and format of the values of the parameter to be monitored.

When the CDMS receives this request, it reports the complete set of actual on-board monitoring parameters.

5.12.2.2 Reporting the Current Parameters out-of-Limit List

Current Parameters Out-of-Limit List Report (12,11)

Telemetry Source Packet, Source Data:

N	Parameter-ID	Parameter Value	Limit Crossed	Transition Time
Unsigned Integer	Enumerated	Deduced	Deduced	Absolute Time

<--- Repeated N times -->

N:

The number of parameters of the Out-of-Limit List that are reported.

Parameter-ID:

The identification of a parameter to be monitored.

Parameter Value:

This gives the value of the telemetry parameter at the actual time (or at which the last nominal sampling of that parameter was conducted). The format and length are uniquely defined for the parameter.

Limit Crossed:

This is the value of the Low Limit, High Limit, or Expected Value which has been crossed/matched. It has the same format and length as the value of the parameter itself.

Transition Time:

The time at which the transition occurred, i.e. the time of the first sample used to elaborate the current checking status.

5.13 Service Type 13: not used

5.14 Service Type 14: Packet Transmission Control

The Packet Transmission Control Service allows for enabling or disabling the transmission of TM Packets of the addressed Application Process or end-user. It is assumed that an end-user manages the associated functions related to the **generation** of TM packets (buffer management etc.) appropriately.

If a process, function, or task has started, or has been resumed, or changes into a state or mode **nominally**, in which it starts to generate data of a certain packet type, it shall **not** be necessary to enable the generation and transmission of that TM packet via this Service 14. Only in exceptional or contingency cases the Packet Transmission Control Service shall be used to control the generation of TM packets by disabling or enabling certain packet types.

Equivalently, the transmission of TM Packets shall **stop automatically** for nominal status changes equivalent to the ones above (stopping, suspending, etc.).

5.14.1 Telecommand Packet Data Field Structure:

5.14.1.1 Controlling the transmission of specified Telemetry Packets

The requests to enable or disable the transmission of telemetry source packets of specified type, sub-type, and structure identifier from the destination Application Process are:

Enable Transmission of Telemetry Packets (14,1)

and

Disable Transmission of Telemetry Packets (14,2)

Telecommand Packet, Application data:

N	Type	Sub-Type	SID
Unsigned Integer (8 bits)	**Enumerated (8 bits)**	**Enumerated (8 bits)**	**Enumerated (8 bits)**

<------------------------------ Repeated N times --------------------------->

N:

The number of TM Packet definitions that follow.

Type:

The Telemetry Packet Service Type.

Sub-Type:

The Telemetry Packet Service Sub-Type for the specified Service Type

.SID:

- **If a Structure ID is defined** for a given TM Packet of a certain Application:
- The Structure Identifier of the TM Packet that shall be enabled /disabled. If the SID is set to **Zero, all packet structures** of a given Type/Subtype are enabled/disabled.

- -If **no** Structure ID is defined for a given TM Packet of a certain Application (i.e. the structure is fixed for all applications), the Structure Identifier field is set to **Zero**.

5.14.1.2 Reporting the list of enabled Telemetry Packets

The request to report the list of telemetry packet types and sub-types from the Application Process with an "Enabled" transmission status is:

Report Enabled Telemetry Packets (14,3)

Telecommand Packet, Application Data: **None**

When this request is received, the enabled telemetry source packet of the addressed Application Process are determined and a report (14,4) is generated.

5.14.2 Telemetry Packet Data Field Structure:

Enabled Telemetry Packets Report (14,4)

Telemetry Source Packet, Source Data:

N	Type	Sub-Type	SID
Unsigned Integer (8 bits)	Enumerated (8 bits)	Unsigned Integer (8 bits)	Enumerated (8 bits)

<------------------- Repeated N times --------------------->

N:

The number of TM Packet definitions that follow.

Type:

The Telemetry Packet Service Type.

Sub-Type:

The Telemetry Packet Service Sub-Type for the specified Service Type.

SID:

- **If a Structure ID is defined** for a given TM Packet of a certain Application:
The Structure Identifier of the TM Packet that is enabled /disabled.
- If **no** Structure ID is defined for a given TM Packet of a certain Application (i.e. the structure is fixed for all applications), the Structure Identifier field is set to **Zero**.

5.15 Service Type 15: On-board Storage and Retrieval

The On-board Storage and Retrieval Service works in support of the Solid State Mass Memory and Non-volatile Memory, which are in charge of storing all TM data (at any time) in several Packet Stores. These recorded TM packets can be transmitted to ground in parallel with realtime data during (daily) periods of contact with a ground station. The functions of the Service 15 are executed by the unit in charge controlling the involved memories i.e. the CDMS TBC.

The selective storing and retrieval of TM packets can be controlled on-board according to a few basic criteria (which are long-term static during nominal operation). Detailed data extraction and ordering shall be done on ground. As all TM packets are time-stamped and uniquely identfied by several other means they need not to be re-formatted during the process of storing and retrieval.

The Packet Stores shall be organised as circular buffers, consequently nominal periodic clearing of the “oldest” part of a Store is not needed. Data are recorded as they arrive at the CDMS, in sequence according to increasing packet time.

If certain TM packet Types/Subtypes are not required to be downlinked for a certain period of time, this shall be controlled by using the capabilities of Service 14 at the sources of packet generation. All TM packets that are generated on-board shall be available for downlinking, and shall be downliked under nominal conditions.

5.15.1 Telecommand Packet Data Field Structure:

5.15.1.1 Controlling the storage in specified packet stores

The requests are:

Enable Storage in Packet Stores (15,1)

Telecommand Packet, Application Data:

Disable Storage in Packet Stores (15,2)

Telecommand Packet, Application data:

N	Store ID
Unsigned Integer	**Enumerated**
Optional	<-------- Repeated N times -------->

N:

The number of packet stores to be controlled. By convention, N = 0 means "all Packet Stores".

Store ID:

An on-board Packet Store is uniquely identified by a "Store ID"., which explicitly indicates, for example, an access path to a physical on-board recording device or file.

When the CDMS receives this request, it starts or stops (depending on whether it is an "enable" or "disable" request) sending the relevant packets to the specified Packet Stores.

5.15.1.2 Modifying the definition of a storage selection criteria

The storage selection definition used by CDMS to send packets for storage in a given Packet Store consists of the identification of the Application Identifier, Type and Subtype of the relevant packets. It is possible to add to (or remove from) a storage selection definition.

The requests to modify the storage selection definition for a specified Packet Store are:

Add Packet Definitions to Storage Selection Definition (15,3)

Telecommand Packet, Application Data:

Remove Packet Definitions from Storage Selection Definition (15,4)

Telecommand Packet, Application Data:

Store ID	N1	Application ID	Type	N2	Sub-Type
Enumerated	**Unsigned Integer**	**Enumerated**	**Enumerated**	**Unsigned Integer**	**Enumerated**
Optional					Repeated N2 times
		<-------------------- Repeated N1 times ------------------>			

Store ID:

The identifier of the Packet Store in which TM packets are stored.

N1:

The number of TM packet definitions that follow.

APID:

The identifier of the unit/application for which TM packets are stored.

Type:

A Telemetry source packet Type.

N2:

The number of Subtype definitions that follow.

Sub-Type:

A Telemetry packet Subtype of the specified Type.

When the CDMS receives this request:

- if **N1 = 0 and Add, (15,3) was commanded**, all types of TM packet from all Application Processes, which are generated on-board, are to be stored in the specified Packet Store;
- if **N1 = 0 and Remove, (15,4) was commanded**, the storing list for the specified Packet Store is cleared completely;

- if **N1 > 0** and **N2 = 0**, the specified Type of Telemetry packet from the Application Process, covering all Subtypes, is added to (if not yet present), or removed from, the list of stored packets of the specified Packet Store (depending on the type of request);
- if **N1 > 0** and **N2 > 0**, the specified Subtypes of Telemetry packets from the Application Process are added to, or removed from, the list of stored packets of the specified Packet Store (depending on the type of request).

Note that if N1 > 1 then there may be a mixture of empty (N2 = 0) and non-empty arrays (N2 > 0).

The current content of the Packet Store is not affected by the request and, if storage is enabled, packets start or stop to be appended to the Packet Store immediately after the command is executed.

5.15.1.3 Reporting a storage selection definition

The request is:

Report Storage Selection Definition (15,5)

Telecommand Packet, Application Data:

Store ID
Enumerated
Optional

Store ID:

The identifier of the Packet Store for which a report is requested.

When this request is received by the CDMS, the storage selection definition for the specified Packet Store is read and a report (15,6) is generated.

5.15.1.4 Downlinking the contents of a packet store (for a slected packet subset)

This Telecommand serves for actually starting the downlinking of stored TM packets. The retrieval and downlinking is controllable in four different ways with this command:

- downlinking the entire contents of a selected Packet Store for a default (relative) time period, or a previously commanded period of time.
- downlinking a selected subset of a selected Packet Store for a default (relative) time period, or a previously commanded period of time.

If a TM packet subset selection is commanded, this selection shall only be effective until the requested downlinking session has finished.The default condition shall be that all data, that are stored in a Packet Store, are downlinked later on.

The request is:

Downlink Packet Store Contents (15,7)

Telecommand Packet, Application Data:

Store ID	N1	Application ID	Type	N2	Sub-Type
Enumerated	Unsigned Integer	Enumerated	Enumerated	Unsigned Integer	Enumerated
Optional					Repeated N2 times
		<-------------------- Repeated N1 times ------------------->			

Store ID:

The identifier of the Packet Store from which TM packets are to be downlinked.

N1:

The number of TM packet definitions that follow.

- if **N1 = 0 (default)**, all stored TM packets for the specified Packet Store is downlinked completely, taking into consideration a previously specified Time Period TC(15,9), or a default time setting;
- if **N1 > 0**, the specified Types/Subtypes of Telemetry packets from the specified Application Processes shall be downlinked

APID:

The identifier of the unit/application for which TM packets are dowmlinked.

Type:

The associated Telemetry source packet Type.

N2:

The number of Subtype definitions that follow.

Sub-Type:

The associated Telemetry packet Subtype of the specified Type.

When this request is received by the CDMS, the contents of the specified Packet Store matching with the specified packet subset is downlinked.

If a certain TM packet definition has no matching entry in the list of stored TM packets for the addressed Packet Store, it shall be ignored and the downlinking of the rest of the data shall be unaffected.

5.15.1.5 Downlinking the contents of a packet store for a specified time period

The request is:

Specify Downlink Time Period (15,9)

Telecommand Packet, Application Data:

Store ID	Time Span	Storage Time 1	Storage Time 2
Enumerated	Enumerated	Absolute Time	Absolute Time
Optional	Optional	<------- Optional ------->	<------- Optional ------->

Store ID:

The identifier of the Packet Store from which TM packets are to be downlinked.

Time Span:

This indicates how the packet range is specified. If Time Span is "All" (value = 0), the full contents of the Packet Store are to be downlinked, otherwise it is the set of packets whose storage times are:

- between Storage Time 1 and Storage Time 2 inclusive, if Time Span is Between" (value = 1);
- less than or equal to Storage Time 1 if Time Span is "Before" (value = 2);
- greater than or equal to Storage Time 1 if Time Span is "After" (value = 3).

Storage Time 1, Storage Time 2:

The absolute time(s) defining the boundary(ies) of the range of packets to be downlinked. Storage Time 1 is present if Time Span is not "All". Storage Time 2 is present if Time Span is "Between".

When this request is received by the CDMS, the time period for the next packet retrieval from Packet Stores is set (different from the default setting). When the downlinking is actually started by executing a Telecommand (15,7) the CDMS downlinks the contents of the specified Packet Store falling within the specified time period. Whatever the value of Time Span, the retrieval ends at the latest when the last packet stored at the time of reception of the request has been downlinked.

5.15.1.6 Deleting the contents of specified packet stores up to a specified storage time

The request is:

Delete Packet Stores Contents up to Specified Storage Time (15,11)

Telecommand Packet, Application Data:

End Time	N	Store ID
Absolute Time	**Unsigned Integer**	**Enumerated**
	Optional	<------ Repeated N times ----->

End Time:

The absolute time defining the upper boundary (inclusive) of the packet range to be deleted.

N:

By convention, N = 0 means "All Packet Stores".

When this request is received by theCDMS, the packets in the specified Packet Stores (all Packet Stores if N = 0) which have a storage time earlier than or equal to the specified time are deleted. The deletion ends at the latest when the last packets stored at the time of reception of the request have been deleted.

While deletion from a Packet Store is in progress, packets sent for storage may be recorded in the freed space of the Packet Store.

5.15.1.7 Reporting packet store catalogues

The request to report the catalogues of selected Packet Stores is:

Report Catalogues for Selected Packet Stores (15,12)

Telecommand Packet, Application Data:

N	Store ID
Unsigned Integer	**Enumerated**
Optional	<-------- Repeated N times -------->

N:

The number of packet stores whose catalogues are to be reported. By convention, N = 0 means "all Packet Stores".

When this request is received by the CDMS, the catalogues for the specified Packet Stores are reported with TM(15,13).

5.15.2 Telemetry Packet Data Field Structure:

As TM packets, that are retrieved from the Packet Stores, are downlinked to ground without any further modification via one or several (dedicated TBC) Virtual Channels TBD, there is no need for the definition of an encapsulating TM packet which belongs to Service 15.

The CDMS has no control over the TM packets that are actually generated by other units/applications (and then stored in one of the Packet Stores). This process may be non-continuous for certain TM packet Types. Therefore only setup parameters and global parameters about Packet Store utilisation are reported.

5.15.2.1 Reporting the Storage Selection Definition (15,6)

The format of this report is:

Storage Selection Definition Report (15,6)

Telemetry Source Packet, Source Data:

Store ID	N1	Application ID	Type	N2	Sub-Type
Enumerated	**Unsigned Integer**	**Enumerated**	**Enumerated**	**Unsigned Integer**	**Enumerated**
Optional					Repeated N2 times
		<-------------------- Repeated N1 times ------------------>			

Store ID:

The identifier of the Packet Store in which TM packets are stored.

N1:

The number of TM packet definitions that follow.

APID:

The identifier of the unit/application for which TM packets are stored.

Type:

The Telemetry source packet Type which is stored for the specified APID.

N2:

The number of Subtype definitions that follow.

Sub-Type:

A Telemetry packet Subtype of the specified Type

5.15.2.2 Reporting the Packet Store Catalogue, (15,13)

The format of this report is:

Packet Store Catalogue Report (15,13)

Telemetry Source Packet, Source Data:

N	Store ID	Storage Time 1
Unsigned Integer	Fixed CharString	Absolute Time

<--------- Repeated N times -------------

Storage Time 2
Absolute Time

---- Repeated N times (cont'd) --->

Percentage Filled	Percentage Downlinked	No. of Packets Stored	No. of Packets Downlinked
Unsigned Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer

-------------------------------- Repeated N times (cont'd) --------------------->

N:

The number of Packet Stores for which the catalogue information is reported.

Store ID:

Packet Store Identifier.

Storage Time 1, Storage Time 2:

The interval in absolute time for which Packet Store Catalogue is reported.

Percentage Filled:

The percentage of the Packet Store occupied by stored packets.

Percentage Downlinked:

The percentage of the Packet Store that has already been downlinked.

No. of Packets Stored:

The total number of packets stored in the Packet Store.

No. of Packets Downlinked:

The number of packets stored in the Packet Store that have already been downlinked.

5.16 Service Type 16: On-board Traffic Management:

For the TM / TC Packet Service 16 no specific packet data structures are defined. The CDMS has to generate the appropriate **Event Packets** in order to report anomalies of the TM / TC Packet routing and distribution function. Control over the generation and transmission of individual packets is executed by utilising Service 14.

5.17 Service Type 17: Test Service

A generic test for all on-board users is an end-to-end "connection test" between the ground and the Application Process.

The function exercised by this Test Service Request is the generation of a corresponding one-shot Service Report by the Application Process. The reception on the ground of the Service Report will serve to confirm that the routes (uplink and downlink) between itself and the Application Process are operational and that the Application Process itself is performing a minimum set of functions (which includes Telecommand processing).

5.17.1 Telecommand Packet Data Field Structure:

The request to perform an end-to-end connection test is:

Perform Connection Test (17,1)

Telecommand Packet, Application Data: **None**

5.17.2 Telemetry Packet Data Field Structure:

On successful receipt of a Telecommand of type TC(17,1), the Application shall respond with a **nominal Successful Command Acceptance report, TM(1,1).**

On unsuccessful receipt of a Telecommand of type TC(17,1), the Application shall respond with a **Failure Command Acceptance report, TM(1,2).**

5.18 Service Type 18: On-board Control Procedures

On-Board Control Procedures (OBCPs) are flight procedures, which are resident on-board of the FIRST or Planck satellite. After activation they are interpreted and executed in the on-board system, e.g. the CDMS, the ACC, and potentially other intelligent on-board users, like instrument control units. They serve for controlling processes, which may be active for an extended period of time and which may involve the (conditional) execution of a (longer) sequence of commands. These on-board Telecommands may affect one or several application processes or functions. More than one on-board unit may be involved.

In order to retain predictable and robust behaviour of the spacecraft and its systems the number of OBCPs shall be kept to a minimum, and the internal structure of each OBCP must be kept simple.

OBCPs have to comply with common requirements on structure, syntax, operability, and maintainability. Further details are specified via the F/P OIRD, AD 3.

5.18.1 Telecommand Packet Data Field Structure:

5.18.1.1 Loading a procedure

The request is:

Load Procedure (18,1)

Telecommand Packet, Application Data:

Procedure ID	Length	Procedure Code
Enumerated	Unsigned Integer	Variable OctetString

Procedure ID:

The unique identification of the procedure that is being loaded.

Length:

The length (in octets) of the procedure code that follows.

Procedure Code:

The code of the procedure (in increasing order of octet).

When this request is received, the addressed unit stores the procedure code and updates its list of loaded on-board procedures accordingly. If the Procedure ID is the same as a procedure currently in the list, the procedure just loaded **replaces** the one that was previously loaded. The status of the loaded procedure is set to “stopped”.

5.18.1.2 Deleting a procedure

The request is:

Delete Procedure (18,2)

Telecommand Packet, Application Data:

Procedure ID
Enumerated

When this request is received, the specified on-board procedure is deleted from the list of loaded on-board procedures and the area occupied by the procedure code is cleared.

The request is ignored if the procedure status is "running" or “suspended”.

5.18.1.3 Starting a procedure

The request is:

Start Procedure (18,3)

Telecommand Packet, Application Data:

Procedure ID	N	Parameter-ID	Value
Enumerated	Unsigned Integer	Enumerated	Deduced
	<----------------- Optional ------------------>		Optional
		<------------ Repeated N times ------------>	

N:

The number of couplets of parameter identifiers and values which follow.

Parameter-ID

The identification of the parameter whose value follows.

Value:

The value of the corresponding parameter (its type will be deduced from the Parameter-ID)

If N is smaller than N_max then the N parameters of the Parameter Field shall be sent to the OBCP, **the other parameters shall retain their current/last value.**

The parameters are used to configure the specific instance of execution of an on-board procedure. When this request is received, the specified on-board procedure is started using (or passing) the specified activation parameters. The procedure status is then “running”.

The request is ignored if the status of the procedure was "running" or “suspended”.

The verification of execution of this “start procedure” request must be reported by using the standard reports of the Telecommand Verification Service (Type 1). For the reporting of the progress of execution of the procedure TC Verification packets or Event packets may be involved.

5.18.1.4 Stopping a procedure

The request is:

Stop Procedure (18,4)

Telecommand Packet, Application Data:

Procedure ID
Enumerated

When this request is received, the specified on-board procedure is stopped. The procedure status is then "stopped".

The request is ignored if the procedure already has the "stopped" status.

5.18.1.5 Suspend a procedure

The request is:

Suspend Procedure (18,5)

Telecommand Packet, Application Data:

Procedure ID	Step ID
Enumerated	Enumerated
	Optional

Step ID:

The Step ID indicates the step at the completion of which the procedure shall be suspended.

When this request is received, the specified on-board procedure is suspended at the completion of the indicated step. The procedure status is then "suspended".

The request is ignored if the procedure status was "stopped" or “suspended”.

5.18.1.6 Resume a procedure

The request is:

Resume Procedure (18,6)

Telecommand Packet, Application Data:

Procedure ID
Enumerated

When this request is received, the specified on-board procedure is resumed after the step where it was previously suspended. The procedure status is then "running".

The request is ignored if the procedure status was "stopped" or “running”.

5.18.1.7 Communicate parameters to a procedure

The request is:

Communicate Parameters to a Procedure (18,7)

Telecommand Packet, Application Data:

Procedure ID	N	Parameter-ID	Value
Enumerated	Unsigned Integer	Enumerated	Deduced
	<-------------------- Optional --------------------->		Optional
		<--------------- Repeated N times --------------->	

N:

The number of couplets of parameter identifiers and values which follow.

Paramete-ID:

The identification of the parameter whose value follows.

Value:

The value of the corresponding parameter (its type will be deduced from the Parameter-ID)

When this request is received, the Application Process passes the parameters to the procedure. The procedure status is unchanged by this action.

The request is ignored if the procedure status was "stopped" or “suspended”.

5.18.1.8 Requesting the list of on-board operations procedures

The request is:

Report List of On-board Control Procedures (18,8)

Telecommand Packet, Application Data: **None**

When this request is received, a report Type (18,9) is generated.

5.18.1.9 Requesting the list of active on-board operations procedures

The request is:

Report List of Active On-board Control Procedures (18,10)

Telecommand Packet, Application Data: **None**

When this request is received, a report Type (18,11) is generated

5.18.2 Telemetry Packet Data Field Structure:

5.18.2.1 Reporting the list of on-board control procedures:

On-board Control Procedures List Report (18,9)

Telemetry Source Packet, Source Data:

NPROC	Procedure ID
Unsigned integer	Enumerated
	<-Repeated NPROC times->

NPROC:

The number of procedures loaded on-board that follow.

5.18.2.2 Reporting the list of active on-board control procedures

Active On-board Control Procedures List Report (18,11)

Telemetry Source Packet, Source Data:

NPROC	Procedure ID
Unsigned integer	Enumerated
	<-Repeated NPROC times->

NPROC:

The number of active on-board procedures that follow.

5.19 Service Type 19: Event / Action Service

The Event / Action Service, implemented in the CDMS, maintains a list of events to be detected, that contains the following information:

- Application Process ID generating the event report;
- Event Report ID (Structure ID);
- Associated action (Telecommand Packet);
- Status of the action - enabled or disabled.

On reception of an event report of Type/Subtype (5,2) and (5,1), the CDMS scans the detection list and if a matching Event Report is detected and the associated action is enabled, the corresponding Telecommand packet is sent to the destination Application Process. (Severe events, which are classified as errors/alarms, Type (5,3), should either be covered by an action, which was triggered by an event of type (5,2) already, or are so unpredictable in nature that no autonomous on-board activity can be initiated.)

In **exceptional cases**, the Event/Action List may be modified by adding or deleting events or activating / deactivating related actions. The related Telecommands may have mission-wide implications and have therefore to be classified as hazardous.

5.19.1 Telecommand Packet Data Field Structure:

5.19.1.1 Adding events to the detection list

The request is:

Add Events to the Detection List (19,1)

Telecommand Packet, Application Data:

Application Process ID	SID	Telecommand Packet
Enumerated	Enumerated	Variable OctetString

Application Process ID:

The identifier of the Application Process generating this event report (TM packet type (5,1) or (5,2)).

SID:

The Structure ID (SID), together with the Application Process ID and TM-Packet type, which is always 5 for this service, identifies the event, which is reported, (and the presence, structure and interpretation of the associated parameter field). Events of different subtype have different Structure IDs.

Telecommand Packet:

The action to be taken (i.e. Telecommand to be sent) when this event report is detected.

When this request is received, the event is added to the detection list and the corresponding action status is set to “disabled”. If a given event is already in the detection list, the action **replaces** the existing one, providing that the current action status is “disabled”. Otherwise, the request to replace that event results in a corresponding error report (Telecommand execution failure, type (1,8)).

5.19.1.2 Deleting events from the detection list

The request is:

Delete Events from the Detection List (19,2)

Telecommand Packet, Application Data:

Application Process ID	SID
Enumerated	Enumerated

Application Process ID, Structure ID:

as for TM packet type (19,1)

When this request is received, the indicated event is deleted from the detection list, provided that the current action status is “disabled”. Otherwise, the corresponding error report must be generated.

5.19.1.3 Clearing the event detection list

The request is:

Clear the Event Detection List (19,3)

Telecommand Packet, Application Data: **None**

When this request is received, all entries in the detection list are cleared.

5.19.1.4 Controlling the actions associated with events

The requests are:

Enable Actions (19,4)

Telecommand Packet, Application Data:

Disable Actions (19,5)

Telecommand Packet, Application Data:

Application Process ID	SID
Enumerated	Enumerated

Application Process ID, Structure ID:

as for TM packet type (19,1)

When this request is received, the action associated with the corresponding event is enabled/disabled.

An error is flagged if an event, whose action is requested for enabling/disabling, is not in the detection list. Changing the enable/disable status of an action shall be executed irrespective of the previous state of the action.

5.19.1.5 Reporting the event detection list

The request is:

Report the Event Detection List (19,6)

Telecommand Packet, Application Data: **None**

When this request is received, a report of type (19,7) is generated.

5.19.2 Telemetry Packet Data Field Structure:

5.19.2.1 Report of the event detection list

Event Detection List Report (19,7)

Telemetry Source Packet, Source Data:

N	Application Process ID	SID	Action Status
Unsigned integer	Enumerated	Enumerated	Enumerated

<--------------------- Repeated N times --------------------->

N:

The number of events in the event detection list.

Application Process ID, Structure ID:

as for TM packet type (19,1)

Action Status:

This indicates the status of the action associated with the event, as follows:

Value = 0 (Disabled)

Value = 1 (Enabled).

5.20 Service Type 20: Information Distribution Service

The CDMS shall be capable of re-routing blocks of information from units or applications to other end-users on-board.These data may serve for checking the status of other units or they may become merged with data like scientific recordings. This can be done by either re-formatting a data field of an Event Report TM packet, which arrives from an application like the AOCS, into a TC packet addressed to another application or unit. Another option is the assembly of parameters, which are available to the CDMS (e.g. from a data pool of housekeeping parameters), into a new TC packet, which is then passed on to its destination.

For the distribution of data to on-board units / applications a special Information TC Packet, type/subtype (20,4), is assigned, with (pre-)defined structures within the on-board software. These TC packets are normally generated after the occurrence of a certain event like finishing of an attitude manoeuvre.

Only Information Telecommands which serve for the purpose of providing information to applications or units shall be disabled or enabled in exceptional cases.

5.20.1 Telecommand Packet Data Field Structure:

5.20.1.1 Controlling the on-board distribution of Telecommand packets

The requests to enable or disable the distribution of Information Telecommand packets of a specified structure to certain destination Application Processes are:

Enable Distribution of Information TC Packets (20,1)

and

Disable Distribution of Information TC Packets (20,2)

Telecommand Packet, Application data:

Application Identifier	SID	Spare
Enumerated (16 bits)	Enumerated (8 bits)	(8 bits)

Application Identifier:

The destination APID of the TC Packet definition that follow.

SID:

The Structure Identifier of the Information Packet that shall be distributed.

5.20.1.2 Reporting the list of distributed Information TC packets

The request to report the list of the Application Processes and the Structure IDs of Information Packets with an "Enabled" distribution status is:

Report Distributed Information Packets (20,3)

Telecommand Packet, Application Data: **None**

When this request is received, the Telecommand packets of the Application Processes, which are enabled for on-board distribution, are determined and a report (20,5) is generated.

5.20.1.3 Distributing Information Telecommands

The structure of Information Telecommands is:

Information Telecommand (20,4)

Telecommand Packet, Application Data:

SID	Parameters
Enumerated	Any

SID:

The SID, together with the Application Process ID and the nature of the packet (packet type / subtype: Information Packet) implicitly identifies the structure of the parameter field.

Parameters:

This field consists of a sequence of values of parameters extracted by the CDMU from one or several sources. The only authorised parameter types are those described in Appendix 6.

5.20.2 Telemetry Packet Data Field Structure:

Distributed Information Packets Report (20,5)

Telemetry Source Packet, Source Data:

N	Application Identifier	SID
Unsigned Integer (8 bits)	Enumerated (16 bits)	Enumerated (8 bits)

<-------------- Repeated N times ---------------->

N:

The number of TC packet definitions that follow.

Application ID, SID:

as for packet type (20,1)

5.21 Service Type 21: Science Data Transfer

This section defines the telemetry packet structures for Science Data. Instrument HK and status data are covered by Telemetry packets of type/subtype TM(3,25) and TM (3,26).

In order to avoid an unnecessarily large number of Application IDs for a single instrument several subtypes of Science TM packets are introduced. These subtypes should be used to identify groups of scientific data of significantly different character or origin (e.g. different detectors). Each subtype can make use of several structure definitions. The individual naming of the Science TM packets can be chosen by the instruments, the number of allowed subtypes is TBC.

5.21.1 Telecommand Packet Data Field Structure:

not applicable

5.21.2 Telemetry Packet Data Field Structure:

5.21.2.1 Reporting Scientific Data

The report packets of the values of a set of scientific parameters are:

Nominal Science Data Report (21,1)

Telemetry Source Packet, Source Data:

Science Type B Data Report (21,2)

Telemetry Source Packet, Source Data:

Diagnostic Science Report (21,3)

Telemetry Source Packet, Source Data:

Auxiliary Science Data Report (21,4)

Telemetry Source Packet, Source Data:

SID	Parameters
Enumerated	Any

SID:

The SID, together with the Application Process ID and the nature of the packet (packet type / subtype) implicitly identifies the structure of the parameter field. Details are to be found in the respective Instrument User Manuals.

Parameters:

This field consists of a sequence of values of instrument (science) parameters that are sampled or prepared once per collection interval followed by a sequence of fixed-length arrays of records.

The only authorised parameter types are those described in Appendix 6.

5.22 Service Type 22: Context Saving Service

On request an addressed user shall provide one TM packet with context and status information. This information shall incorporate all necessary data that is needed to re-activate a certain unit (after switch-off) in a way that the same basic (hardware) configuration is established as before.

After powering a certain unit, which can act as a Packet end user, it has to conduct a self-test and has to establish among others the basic TC / TM processing services. **One Context TC packet** can then be routed to that unit, which may be used to activate subsequent units of that end user and to set several basic parameters. The unit (or group of units belonging e.g. to an instrument) should be in an Idle or Stand by Mode thereafter, ready to accept specific Telecommands with parameters for configuring that unit into one of its nominal operational modes.

5.22.1 Telecommand Packet Data Field Structure:

TBD

5.22.2 Telemetry Packet Data Field Structure:

TBD

Appendix 1 CONVENTIONS

A1.1 Bit Numbering Conventions

The following convention shall be used to identify each bit in an N-bit field :

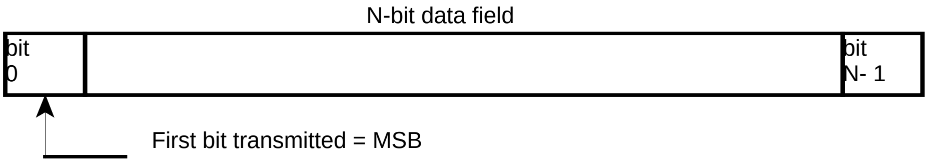


Figure A1.1 Bit numbering

1) The first bit in the field (starting from the left) is defined to be “Bit 0” and will be represented as the left most justified bit in a figure. The next bit is called “Bit 1”, and so on, up to “Bit N-1”, the bits being represented in this order from left to right in a figure.
2) If the N-Bit field is to be interpreted as “Unsigned Integer” value, Bit 0 is the MSB and Bit N-1 is the LSB.
3) If the N-Bit field is to be interpreted as “Signed Integer” value, Bit 0 indicates the sign with Bit 0 = 0 corresponding to a positive number and Bit 0 = 1 corresponding to a negative number.
4) Adjacent groups of bits are described in terms of octets and words.
5) Octet = 1 byte = 8 bits (1 word = 2 octets = 16 bits).

A1.2 Field Alignment Conventions

The following convention shall be used to construct packet parameter fields:

1) Parameters with a length longer or equal 16 bits shall be word aligned, i.e. the LSB shall coincide with the word boundary.
2) Parameters with a length shorter than 16 bits shall not be allowed to span over word boundaries.
3) Parameters with a length shorter than 16 bits shall be right-adjusted within the occupied 16-bit word, leaving any required padding-bits in the most significant bits of the 16-bit word.
4) If more than one parameter is held in a single word the parameters shall be right adjusted.

A1.3 Packet Numbering Conventions

Packet class and function is provided by packet type and packet subtype, included in the data field header of the packet.

The Packet Type numbering scheme is devised to provide correlation between TC packets and the resulting TM packets and is therefore non-contiguous: there are cases where for a certain TC type, there is no corresponding TM type. Appendix 2 provides a complete cross-reference table down to sub-type level.

To make identification simpler, service type and subtype are represented by two numbers, separated by a comma, for example, TC (1,1) is a Telecommand packet type 1, subtype 1 and TM (1,2) is a telemetry packet type 1, subtype 2. Subtype numbers within a service shall be unique.

Appendix 2 TM/ TC PACKET TYPES AND SUBTYPES FOR FIRST/ PLANCK

Table A-2.1 TM/ TC Packet Types/Subtypes Summary

TC-Packet Type/Subtype	TM-Packet Type/Subtype	TM/TC Packet Name
	TM (1,1)	Telecommand Acceptance Report - Success
	TM (1,2)	Telecommand Acceptance Report - Failure
	TM (1,3)	Telecommand Execution Report - Started
	TM (1,6)	Telecommand Execution Report - Progress
	TM (1,7)	Telecommand Execution Report - Completed
	TM (1,8)	Telecommand Execution Report - Failure
TC (2,1)		Distribute Pulse Commands
TC (2,2)		Distribute Register Load Commands
TC (2,3)		Distribute CPDU Commands
	TM (3,25)	Housekeeping Parameter Report
	TM (3,26)	Diagnostic Parameter Report
	TM (5,1)	Event Report
	TM (5,2)	Exception Report
	TM (5,3)	Error/Alarm Report
TC (6,2)		Load Memory Using Absolute Addresses
TC (6,5)		Dump Memory Using Absolute Addresses
	TM (6,6)	Memory Dump, Absolute Addresses
TC (6,9)		Check Memory Using Absolute Addresses
	TM (6,10)	Memory Check Repoprt, Absolute Addresses
TC (7,1)		Start Task
TC (7,2)		Stop Task
TC (7,3)		Suspend Task
TC (7,4)		Resume Task
TC (8,1)		Start Function
TC (8,2)		Stop Function

TC (8,3)		Perform Activity of Function
TC (8,4)		Load Function Parameters
TC (9,3)		Synchronise User
TC (9,4)		Enable Time Synchronisation
TC (9,5)		Time Code
TC (9,6)		Verify User Time
TC (9,7)		Enable Time Verification
	TM (9,8)	Central Time Reference
	TM (9,9)	Time Verification Report
TC (11,1)		Enable Release of Telecommands
TC (11,2)		Disable Release of Telecommands
TC (11,3)		Reset Command Schedule
TC (11,4)		Insert MTL-Telecommands in Command Schedule
TC (11,5)		Delete MTL-Telecommands
TC (11,6)		Delete MTL-Telecommands over Time Period
TC (11,7)		Time-shift Telecommands
TC (11,8)		Time-shift Telecommands over Time Period
TC (11,9)		Report Command Schedule in Detailed Form
	TM (11,10)	Detailed Schedule Report
TC (11,12)		Report Command Schedule in Summary Form
	TM (11,12)	Summary Schedule Report
TC (11,14)		Report Schedule in Detailed Form over Time Period
TC (12,1)		Enable Monitoring of Parameters
TC (12,2)		Disable Monitoring of Parameters
TC (12,4)		Clear Monitoring List
TC (12,5)		Modify Monitoring List
TC (12,6)		Delete Parameters from Monitoring List
TC (12,8)		Report Current Monitoring List
	TM (12,9)	Current Monitoring List Report
TC (12,10)		Report Current Parameters Out-of-Limit List
	TM (12,11)	Current Parameters Out-of-Limit List Report
TC (14,1)		Enable Transmission of Telemetry Packets
TC (14,2)		Disable Transmission of Telemetry Packets
TC (14,3)		Report Enabled Telemetry Packets

	TM (14,4)	Enabled Telemetry Packets Report
TC (15,1)		Enable Storage in Packet Stores
TC (15,2)		Disable Storage in Packet Stores
TC (15,3)		Add Packet Definitions to Packet Store
TC (15,4)		Remove Packet Definitions to Packet Store
TC (15,5)		Report Storage Selection Definition
	TM (15,6)	Storage Selection Definition Report
TC (15,7)		Downlink Packet Store Contents
TC (15,9)		Specify Downlink Time Period
TC (15,11)		Delete Packet Stores Contents up to Specified Time
TC (15,12)		Report Catalogues for Selected Packet Stores
	TM (15,13)	Packet Stores Catalogue Report
TC (17,1)		Perform Connection Test
TC (18,1)		Load Procedure
TC (18,2)		Delete Procedure
TC (18,3)		Start Procedure
TC (18,4)		Stop Procedure
TC (18,5)		Suspend Procedure
TC (18,6)		Resume Procedure
TC (18,7)		Communicate Parameters to a Procedure
TC (18,8)		Report List of On-board Control Procedures
	TM (18,9)	On-board Control Procedures List Report
TC (18,10)		Report List of Active OBCPs
	TM (18,11)	Active OBCPs List Report
TC (19,1)		Add Events to Detection List
TC (19,2)		Delete Events from Detection List
TC (19,3)		Clear the Event Detection List
TC (19,4)		Enable Actions
TC (19,5)		Disable Actions
TC (19,6)		Report the Event Detection List
	TM (19,7)	Event Detection List Report
TC (20,1)		Enable Distribution of Information TC Packets
TC (20,2)		Disable Distribution of Information TC Packets
TC (20,3)		Report Distributed Information Packets

TC (20,4)		Information Telecommand
	TM (20,5)	Distributed Information Packets Report
	TM (21,1)	Nominal Science Data Report
	TM (21,2)	Science Type B Data Report
	TM (21,3)	Diagnostic Science Data Report
	TM (21,4)	Auxiliary Science Data Report
TC (22,x)		Context Saving, TBD
	TM (22,x)	TBD

Appendix 3 APPLICATION PROCESS ID ASSIGNMENT

Table A-3.1 below lists the allocation of APIDs to the various **on-board users** within the FIRST / Planck Project. The assignment of Application IDs is managed by the FIRST / Planck Project Office. A common structure of APID-allocation shall be adopted for all users for standardised functions.

Certain ranges of APIDs are allocated for instruments according to their specific needs. Additionally, one range of APIDs is reserved for **testing and EGSE-related messages**. This range shall not be used for the addressing of any on-board unit or application. It may be utilised by several groups in parallel during their unit-level test activities before integration of these units into the spacecrafts.

The Application Process ID (APID) is structured into two fields:

The most significant bits of each APID form a base address, which identifies in general terms the user or process generating the TM packet (or which the TC packet is addressed to) in a unique way. Each instrument and subsystem (TBC) on-board is associated to one base APID.

The least significant 3 bits within the APID form a field, which identifies different categories of TM packets. This allows to identify a certain service type of a packet without decoding the Data Field Header of a packet, among others in order to support routing of different types of data via different Virtual Channels.

For issue 1 of this document, all APID allocations to S/C units and S/Ss are to be considered to be TBC.

Table A-3.1 Application Process ID Assignment

APID		**assigned to:**
decimal	hex.	
0	0h	TM Time Source Packets
16	10h	CDMS, Telecommands, TC-Verification, Events
18		- " - , Periodic HK-Packets
64	40h	OBCP, Telecommands, TC-Verification, Events
128	80h	System, Telecommands, TC-Verification, Events
130		- " - , Periodic HK-Packets
512	200h	AOCS, Telecommands, TC-Verification, Events
514		- " - , Periodic HK-Packets
576	240h	EPS, Telecommands, TC-Verification, Events
578		- " - , Periodic HK-Packets
640	280h	TCS, Telecommands, TC-Verification, Events
642		- " - , Periodic HK-Packets
768	300h	TT&C, Telecommands, TC-Verification, Events
770		- " - , Periodic HK-Packets
1024	400h	HIFI, Telecommands, TC-Verification, Events
1026		- " - , Periodic HK-Packets

1028 to		- “ - , Science and Diagnostic TM-Packets
1031		
1152	480h	PACS, Telecommands, TC-Verification, Events
1154		- “ - , Periodic HK-Packets
1156 to		- “ - , Science and Diagnostic TM-Packets
1159		
1280	500h	SPIRE, Telecommands, TC-Verification, Events
1282		- “ - , Periodic HK-Packets
1284 to		- “ - , Science and Diagnostic TM-Packets
1287		
1408	580h	HFI, Telecommands, TC-Verification, Events
1410		- “ - , Periodic HK-Packets
1412 to		- “ - , Science and Diagnostic TM-Packets
1415		
1536	600h	LFI, Telecommands, TC-Verification, Events
1538		- “ - , Periodic HK-Packets
1540 to		- “ - , Science and Diagnostic TM-Packets
1543		
1664	680h	SCS, Telecommands, TC-Verification, Events
1666		- “ - , Periodic HK-Packets
2016 to	7E0h	SCOE / EGSE reserved
2046		
2047	7FFh	Idle TM-Packets

Appendix 4 THE CHECKSUM ALGORITHMS

A4.1 Cyclic Redundancy Check Code Specification

The Packet Error Control Field is a 16-bit field, which occupies the two trailing octets of a TC Packet.

The purpose of this field is to provide a capability for detecting errors which may have been introduced into the frame by the lower protocol layers during the transmission process and may have remained undetected.

The standard error detection encoding/decoding procedure, which is described in detail in the following paragraphs, produces a 16 bit Packet Check Sequence (PCS) which is placed in the Packet Error Control Field.

This code is intended only for error detection purpose and shall not be used for error correction.

The characteristics of the PCS are those of a cyclic redundancy check code (CRC) and are generally expressed as follows:

a) The generator polynomial is $G(x) = X^{16} + X^{12} + X^{5} + 1$

b) Both encoder and decoder are initialised to the “all-ones” state for each Packet.

c) PCS generation is performed over the entire Packet including the Packet Header less the final 16-bit PCS. (In case the CRC checksum algorithm is applied for the checking of memory data blocks, the checksum is generated over the entire contents of this block.)

d) The code has the following capabilities when applied to an encoded block of less than 32768 bits (2^{15} bits) :

- All error sequences composed of an odd number of bit errors will be detected
- All error sequences containing two bit errors anywhere in the coded block will be detected
- If a random error sequence containing an even number of bit errors (greater than or equal to four) occurs within the block, the probability that the error will be undetected in approximately 2^{-15} (or 3×10^{-5}).
- All single error bursts spanning 16 bits or less will be detected provided no other errors occur within the block.

A4.2 Encoding Procedure

The encoding procedure accepts an (n-16)-bit message and generates a systematic binary (n, n-16) block code by appending a 16-bit Packet Check Sequence (PCS) as the final 16 bits of the block. This PCS is inserted into the Packet Error Control Field. The equation for PCS is:

$PCS = [X^{16}. M (X) + X^{(n-16)}. L (X)]$ MODULO $G (X)$

Where

M (X) is the (n-16)-bit message to be encoded expressed as a polynomial with binary coefficients, n being the number of bits in the encoded message (i.e. the number of bits in the complete Packet).

L (X) is the pre-setting polynomial given by:

$L (X) = S^{15}_{i=0} Xi$ (all “1” polynomial of order 15)

G (X) is the CCITT Recommendation V.41 generating polynomial given by:

$G(X) = X^{16} + X^{12} + X^{5} + 1$

Where + is the modulo 2 addition operator (exclusive OR)

Note that the encoding procedure differs from that of a conventional cyclic block encoding operation in that the $X^{(n-16)}$. L (X) term has the effect of presenting the shift register to an all ones state (rather than a conventional all zeros state) prior to encoding.

A4.3 Decoding Procedure

The error detection syndrome, S (X) is given by

$S(X) = (X^{16} . C^*(X) + X^{n} . L(X)$ MODULO G (X)

Where C*(X) is the received block in polynomial form.

S (X) is the syndrome polynomial which will be zero if no error has been detected.

A4.4 Verification of Compliance

The binary sequences defined in this section are provided to the designers of packet systems as samples for testing and verification of a specific CRC error detection implementation.

All data are given in hexadecimal notation. For a given field (data or CRC), the left most hexadecimal character contains the most significant bit (i.e. bit 0 of the CCSDS convention).

DATA	Packet Check Sequence (CRC)
00 00	1D 0F
00 00 00	CC 9C
AB CD EF 01	04 A2
14 56 F8 9A 00 01	7F D5

A4.5 Possible realisations of Packet Check Sequence Encoders/Decoders

CRC encoders and decoders can be implemented in hardware as well as in software. A possible H/W implementation of an encoder and decoder is described in **[AD-1]** and **[AD-2]**. A C-language version is provided in **[RD-1]**.

Appendix 5 ACRONYMS AND GLOSSARY OF TERMS

A5.1 Acronyms

Acronym	Description
ACK	Acknowledgement
AD	Applicable Document
AOCS	Attitude & Orbit Control Subsystem (ACMS)
APID	Application Process Identifier
CDMS	Command and Data Management System, On-board Data Handling System (OBDH)
CCSDS	Consultative Committee for Space Data Systems
CPDU	Command Pulse Distribution Unit
CRC	Cyclic Redundancy Check
CUC	CCSDS Unsegmented Time Code
CTR	Central Time Reference
DEC, dec.	Decimal
DMS	Data Management System
EEPROM	Electrically Erasable PROM
EGSE	Electrical Ground Support Equipment Check
EID	Event Identifier
EPS	Electrical Power Subsystem
ESA	European Space Agency
ESOC	European Space Operations Centre
FID	Function Identifier
FIFO	First In First Out
HEX, hex.	Hexadecimal
HFI	High Frequency Instrument (Planck)
HIFI	Heterodyne Instrument for FIRST
HK	Housekeeping
HL	High Limit
ICD	Interface Control Document
ID	Identifier
ISO	International Standards Organisation

Acronym	Description
LFI	Low Frequency Instrument (Planck)
LGA	Low Gain Antenna
LL	Low Limit
LSB	Least Significant Bit
MID	Memory Identifier
MINT	Monitoring Interval
MOC	Mission Operations Centre
MSB	Most Significant Bit
MSSW	Mission Specific Software
MTL	Mission Time Line
N/A, n.a.	Not Applicable
OBCP	On-Board Control Procedure
OCF	Operational Control Field
PACS	Photoconductor Array Camera and Spectrometer (FIRST)
PB	Play Back
PEC	Packet Error Control
PID	Parameter Identifier
PROM	Programmable Read Only Memory
PSS	Procedures, Specifications and Standards
PUS	Packet Utilisation Standard
RCS	Reaction Control Subsystem
RAM	Random Access Memory
RF	Radio Frequency
RL	Register Load
RLA	Register Load Address
ROM	Read Only Memory
RSS	Root Sum Square
RT	Real Time
RTU	Remote Terminal Unit
SASW	Standard Application Software
S/C	Spacecraft
SCET	Spacecraft Elapsed Time
SCOE	Special Check Out Equipment

Acronym	Description
SCS	Sorption Cooler Subsystem (Planck)
SDU	Service Data Unit
SID	Structure Identifier
SOC	Science Operations Centre
SPIRE	Spectral and Photometric Imaging Receiver (FIRST)
S/S	Subsystem (of Spacecraft)
SSMM	Solid State Mass Memory / Solid State Recorder
TAI	Temps Atomique International
TBC	To Be Confirmed
TBD	To Be Defined
TBM	Time Broadcast Message
TBW	To Be Written
TC	Telecommand
TCS	Thermal Control Subsystem
TID	Task Identifier
TM	Telemetry
TT&C	(RF) Tracking, Telemetry, and Command
UTC	Universal Time Coordinate(d)
VC	Virtual Channel

A5.2 Glossary of Terms

Application (Process) A continuous series of actions to bring about a result for a user. Such process may be on-board (or on ground in special cases). Usually an application process can be associated with a unit, subsystem, or instrument. An Application can receive TC packets and/or generate TM packets.

Application data Data associated with an (on-board) application process, encapsulated in a TC or TM Packet.

Application Process ID An 11 bit address field. The APID of a TM packet identifies the application process which generates the packet. The APID of a TC packet identifies the application process which receives the packet. An APID is unique across the system (space and ground segment of a certain spacecraft).

Function A definite amount of actions to bring about a result for a user. A Function has a control/command interface One or more Functions may be active simultaneously within a application process.

Function ID Function Identifier, identifies a function and defines the structure of the parameter field in the packet. The same FID may be used by different APIDs.

Memory ID Memory Identifier, identifies a memory within an application or unit, which can be addressed individually . The same MID may be used by different APIDs.

Mission Timeline Sequence of time-tagged Telecommands, which are stored in mass memory of the CDMS and are used to control the nominal operation of the satellite and its instruments for up to 48 hours.

Non-intelligent end user On-board user which does not decode TC packets or encode TM packets.

On-Board Control Procedure On-board resident sequence of commands, which can be activated for interpretation and execution in order to control one or more units/ applications or functions.

Packet end user On-board user (unit, subsystem, instrument), which decodes TC packets and encodes TM packets. A packet end user may have more than one application process.

(Functional) Parameter Variable that controls the result of a command, task or process or delivers the value of a measurement, status acquisition, or data processing.

Parameter ID Parameter Identifier that uniquely identifies a parameter across the system. The same PID may not be used by different APIDs.

Process See application process.

Register A set of binary memory cells, fixed by design, to which data can be written and/or data can be read from.

Structure ID Structure Identifier, defines the structure of the parameter field in the packet. The same SID may be used by different APIDs.

Source Data Data generated by an on-board application process, encapsulated in a TM packet.

Task A definite amount of actions to bring about a result for a higher level process like a Function or OBCP. One or more tasks may be active simultaneously within a application process.

Task ID Task identifier, identifies a task within an application. The same TID may be used by different APIDs.

User See Packet end user

Appendix 6 PARAMETER TYPES AND STRUCTURES

A6.1 Introduction

This appendix defines the terminology to be used for any packet description referred to in section 3 or 4.

Each field in a Telecommand or telemetry packet described in this document is designed to hold a parameter value. Each parameter field has a type, defining the set of values that can be assigned to the parameter. The parameter types are defined below.

This appendix defines the physical encoding rules for each type, i.e. the permitted lengths of the parameter fields and the internal format used to encode values. This appendix does not define the conversion of data parameters into physical or engineering units or user messages.

When defining Telecommand and telemetry packets only parameter types defined in this document shall be allowed.

A6.2 Encoding formats of parameter types

The parameter type defines the range of possible parameter values. A given parameter type can vary in format and length. Each combination of parameter type and encoding format has an associated parameter code, which defines the type and its physical encoding.

The parameter code shall be used whenever a definition of a parameter field is required. The parameter codes shall be applicable to both Telecommand and telemetry data.

The parameter code PC, is defined as follows:

Parameter Type Code (PTC)	Parameter Format Code (PFC)
4 bits	4 bits

The parameter code is written as (PTC, PFC) in the tables below.

A6.3 Parameter type definitions

A6.3.1 Boolean

Parameter Type	**PTC**	**PFC**	**Length**	**Value/Range**
Boolean	1	0	1 bit	0 = false, 1 = true

A6.3.2 Enumerated Parameter

Enumerated parameters are parameters with distinct integer values only involved in logical operations (as opposed to numeric operations). The values that such a parameter can take are discrete and un-ordered. An error code is a typical example.

Parameter Type	PTC	PFC	Length
Enumerated Parameter	2	1	1 bit
	2	2	2 bits
	2	3	3 bits
	2	4	4 bits
	2	5	5 bits
	2	6	6 bits
	2	7	7 bits
	2	8	8 bits
	2	12	12 bits
	2	16	16 bits

A6.3.3 Unsigned Integer

Parameter Type	PTC	PFC	Length	Value/Range
Unsigned Integer	3	0	4 bits	{0...15}
	3	1	5 bits	{0...31}
	3	2	6 bits	{0...63}
	3	3	7 bits	{0...127}
	3	4	8 bits	{0...255}
	3	5	9 bits	{0...511}
	3	6	10 bits	{0...1023}
	3	7	11 bits	{0...2047}
	3	8	12 bits	{0...4095}
	3	9	13 bits	{0...8191}
	3	10	14 bits	{0...16383}
	3	11	15 bits	{0...32767}
	3	12	2 octets	{0...65536}
	3	13	3 octets	{0...2exp24 - 1}
	3	14	4 octets	{0...2exp32 - 1}

A6.3.4 Signed Integer

Parameter Type	PTC	PFC	Length	Value/Range
Signed Integer	4	0	4 bits	{-8...7}
	4	1	5 bits	{-16...15}
	4	2	6 bits	{-32...31}
	4	3	7 bits	{-64...63}
	4	4	8 bits	{-128...127}
	4	5	9 bits	{-256...255}
	4	6	10 bits	{-512...511}
	4	7	11 bits	{-1024...1023}
	4	8	12 bits	{-2048...2047}
	4	9	13 bits	{-4096...4095}
	4	10	14 bits	{-8192...8191}
	4	11	15 bits	{-16384...16383}
	4	12	2 octets	{-32768...32767}
	4	13	3 octets	{-2exp23...2exp23 - 1}
	4	14	4 octets	{-2exp31...2exp31 - 1}

A6.3.5 Real

Parameter Type	PTC	PFC	Length	Sign	Exponent	Fraction
Real	5	1	4 octets	bit 0	bit 1 - bit 8	bit 9 - bit 31
	5	2	4 octets	N/A	bit24 -bit31	bit 0 - bit 23

Two formats for real numbers shall be allowed:

PC(5,1): 32-bit single format according to ANSI/IEEE Std 754-1985.
(used also for internal parameter of DSP21020)

PC(5,2) 32-bit single format according to MIL-STD-1750-A
used also for interface of DSP21020 with AIU/AOCMS-SW).

PC(5,1):

Fraction and exponent shall be interpreted as unsigned integers and their values inserted in the formulas given below to determine the value of the parameter.

Fraction is added to a binary 1 to generate the mantissa. To increase the precision of the real number the first bit of the mantissa is assumed to be ‘1’. This is possible since the mantissa should always be normalised, i.e. the mantissa is left shifted and the exponent decremented until a ‘1’ is found in the most significant bit. The resulting range of the mantissa is $(1.0000...)_{DEC}$ to $(1.9999...)_{DEC}$.

To increase the range at the small end, fraction is added to a binary 0 as the binary fraction when the exponent equals zero.

The following rules shall apply to the interpretation of parameters of type real:

Exponent	Fraction	Value
255	<>0	not a defined number
255	0	(-1)^(sign) * infinity
< >255, <>0	any	(-1)^(sign) * 2^(exponent-127) * (1.fraction)
0	<>0	(-1)^(sign) * 2^(-126) * (0.fraction)
0	0	0

Examples:

1/ sign = 0
exponent = 127
fraction = 110 0000 0000 0000 0000 0000
value = $-1^0 * 2^{(127-127)} * (1.110\ 0000\ 0000\ 0000\ 0000\ 0000)_{BIN} = 1.75_{DEC}$

2/ sign = 0
exponent = 0
fraction = 110 0000 0000 0000 0000 0000
value = $-1^0 * 2^{(0-126)} * (0.110\ 0000\ 0000\ 0000\ 0000\ 0000)_{BIN} = 2^{-126} * 0.75_{DEC}$

PC(5,2):

The fraction and the mantissa are to be interpreted as two’s complement numbers. The range is form ± 1.5*10^-39 to ± 1.7*10^-38.

A6.3.6 Time

Parameter Type	PTC	PFC	Length	Coarse	Fine	Format
Time	9	17	6 octets	4	2	CUC

The format is CCSDS Unsegmented Time Code, CUC, as defined in **[AD1]** without P-field.

A packet end user that cannot provide in telemetry a time synchronised with CRT shall flag this by setting the MSB of the Time Field to “1”. In this case the meaning of the rest of the field is user specific.

Appendix 7 STANDARD SPACECRAFT TIME SOURCE PACKET

The Standard Spacecraft Time Source Packet shall be used to transport the regular Spacecraft Elapsed Time samples to ground for time correlation with UTC by the ground segment during periods of ground contact. Its structure is defined in **[AD-1]** and it is shown in figure A7-1 below.

SOURCE PACKET HEADER (48 bits)							**PACKET DATA FIELD (64 bits)**		
PACKET ID				PACKET SEQUENCE CONTROL		PACKET LENGTH	S-FIELD	P-FIELD	T- FIELD
Version Number	Type	Data Field Header Flag	Application Process ID	Segmentation Flags	Source Sequence Count				
3	1	1	11	2	14				
16				16		16	8	8	48

Figure A7-1 Standard Spacecraft Time Source Packet Fields

The time carried by the T-field of the packet shall relate to the instant of occurrence of the leading edge of the first bit of the attached synchronisation marker of the telemetry transfer frame of virtual channel “0” with a virtual channel frame count of “0”.

The field contents of the Standard Spacecraft Time Source Packet header and data field are specified below:

Packet ID:

Version Number:

The version number must be set 000_{BIN}.

Type :

The type must be set to zero.

Data Field Header Flag :

The data field header flag must be set to zero. No data field!

Application Process ID :

The Application process ID shall be set to all zeros.

Sequence Control:

Segmentation [Grouping] Flags:

The segmentation flags must be set to “11”.

Source Sequence Count:

The source sequence count of the time packet must be incremented by 1 whenever the source releases a packet. Ideally, this counter should never re-initialise, however, under no circumstances shall it “short-cycle” (i.e. have a discontinuity other than to a value zero).
The counter wraps around from 2^{14} -1 to zero.

Packet Length:

The packet length field specifies the number of octets contained within the Packet Data Field. The number is an unsigned integer “C” where:

C = (Number of octets in Packet Data Field) - 1

In this case, the number of octets is eight (i.e. C=7).

It should be noted that the actual length of the entire Standard Spacecraft Time Source Packet, including the Packet Header, is 6 octets longer.

S-Field:

Bits 0 through 3 are not used and must be set to zeros.
Bits 4 through 7 shall be set to a value corresponding to the generation frequency of a Standard time packet

P-Field:

Must be set to "00101110" to indicate that the following time format consists of 4 coarse time octets and 2 fine time octets.

T-Field:

This field will contain the Spacecraft Elapsed Time, consistent with the CCSDS Unsegmented Time Code (CUC) format, This field is synchronised to TAI after setting of the on-board time.

Bits 0 through 31 must contain the coarse Spacecraft Elapsed Time as an unsegmented binary count of seconds.
Bits 32 through 47 must contain the fine Spacecraft Elapsed Time as an unsegmented binary power of subseconds.

Appendix 8 IDLE PACKET STRUCTURE

The idle packet will be used to fill the telemetry transfer frame when a frame has to be transmitted and an insufficient number of source packets are available to complete the transfer frame. This may be the case when the source data rate is low compared to the frame period. Its structure is as shown in figure A8-1 below.

SOURCE PACKET HEADER (48 bits)							PACKET DATA FIELD (VARIABLE)
PACKET ID				PACKET SEQUENCE CONTROL		PACKET LENGTH	FILLER PATTERN
Version Number	Type	Data Field Header Flag	Application Process ID	Segmentation Flags	Source Sequence Count		
3	1	1	11	2	14		
16				16		16	Variable

Figure A8-1 Idle Packet Fields

The field contents of the Idle Packet header and data field are specified below:

Packet ID:

Version Number :

The version number must be set to 000_{BIN}.

Type :

The type must be set to zero.

Data Field Header Flag:

The data field header flag must be set to zero.

Application Process ID:

The Application process ID must be set to all ones (7FFhex.).

Sequence Control:

Segmentation [Grouping] Flags :

The segmentation flags must be set to “11”.

Source Sequence Count :

The source sequence count of the idle packet must be incremented by 1 whenever the source releases a packet. Ideally, this counter should never re-initialise, however, under no circumstances shall it “short-cycle” (i.e. have a discontinuity other than to a value zero). The counter wraps around from 2^{14} -1 to zero.

Packet Length:

The packet length field specifies the number of octets contained within the Packet Data Field. The number is an unsigned integer “C” where:

C = (Number of octets in Packet Data Field) - 1

The length of the packet may be chosen freely by the user. It should be noted that the actual length of the entire Idle Packet, including the Packet Header, is 6 octets longer.

Filler Pattern:

The content of the Idle Packet data field shall be random data.

Appendix 9 Satellite Data Bus Protocol Specification

Satellite Data Bus Protocol Specification

The specification provided in **Appendix 9** of this ICD, defines the basic electrical and data protocol requirements for instruments and other units, which are connected via a **Serial Data Bus according to MIL-STD-1553B** to the Data Management Subsystem.

Because it addresses requirements, which are associated with the three lowermost layers of the on-board communication infrastructure (Physical, Data Link, and Transfer Layer), these requirements are treated within this ICD as a self-standing set. They are largely independent from the requirements of the Application Layer (i.e. the specifications for TM/TC packet structures in the main body of this ICD).

Consequently the Appendix 9 is written with its own paragraph numbers, figure and table numbers, and its own list of applicable and reference documents. If references are made to certain paragraphs etc., or to applicable documentation, they refer to the corresponding items of Appendix 9 only.

Table of Contents

1. Introduction

1.1. Scope of this Specification

The environment for the Satellite Data Bus Protocol (SDBP) is a spacecraft carrying several instruments, which are connected via a serial data bus according to MIL-STD-1553B to the Data Management Subsystem.

A common interface specification is established within this document, which is binding for all units on-board, that exchange data across the data bus. This document is the basis for covering all necessary aspects, which enables a party to design, develop, and test this interface as part of a certain unit in a way, that it communicates properly together with other units of the spacecraft after integration.

This specification is made applicable for the corresponding S/C interfaces of the experiments, for S/C Service Module units the corresponding details of the data bus and the protocol are to be agreed upon during Phase B activities. As all detailed I/F requirements for these units are naturally TBD at the time of the release of the FIRST/Planck ITT, the protocol specification below must be incomplete and has placeholder requirements in several cases.

1.2. Applicable Documents

AD 1 MIL-STD-1553B, Notice 2, 8.9.1986
DIGITAL TIME DIVISION COMMAND/RESPONSE MULTIPLEX DATA BUS

AD 2 MIL-HDBK-1553A, Multiplex Applications Handbook
Department of Defense, 1.11.1988

1.3. Reference Documents

RD 1 FIRST / PLANCK Instrument Interface Document – IID Part A
SCI-PT-IIDA-04624, Issue 1

RD 2 FIRST / PLANCK Packet Structure Interface Control Document PS-ICD
SCI-PT-ICD-07527, Issue 1

RD 3 ESA Packet Telemetry Standard
PSS-04-106, Issue 1, 1988

RD 4 ESA Packet Telecommand Standard
PSS-04-107, Issue 2, 1992

1.4. Acronyms

AD	Applicable Document
BC	Bus Controller
CCSDS	Consultative Committee for Space Data Systems
CDMS	Command and Data Management Subsystem
ESA	European Space Agency
FDIR	Fault Detection, Isolation and Recovery
H/K	House Keeping
H/W	Hardware
I/F	Interface
ICD	Interface Control Document
kbps	kilo bit per second
LSB	Least Significant Bit
MSB	Most Significant Bit
PDU	Power Distribution Unit
PSS	Procedures, Specifications and Standards
RD	Reference Document
RT	Remote Terminal
SA	Subaddress
S/C	Spacecraft
SDBP	Satellite Data Bus Protocol
TBC	To Be Confirmed
TBD	To Be Determined
TC	Telecommand
TM	Telemetry

1.5. Bit and Field Numbering Conventions

- Bit numbering

    - The relation between a 20 bit MIL-Bus word and the Bit Position numbering is given below. For all Data Fields the MSBs correspond to Bit Position 00.

Bit times :

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20

Bit # (Bit Position) :

S	S	S	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	P

The most significant bit shall be transmitted first with the less significant bits following in descending order of value in the data word.

See AD 1, paragraph §4.3.

1.6. General Conventions

1.6.1. Subaddresses

Subaddress naming in the following chapters is done by using this arrangement:

Subaddress + Number (used range 1-30) + **T**ransmit or **R**eceive

Example :
SA 10 R - has the meaning: Subaddress Number 10 Receive

1.6.2. Meaning of RT/ Instrument

The meaning of RT (Remote Terminal) / Instrument, which is in the MIL STD 1553B only the bus I/F, encloses in the following chapters the RT, and also the attached subsystems, host controllers and so on. Exceptions are explicitly mentioned.

1.6.3. Meaning of BC

The meaning of BC (Bus Controller), which is in MIL STD 1553B only the bus master, encloses in the following chapters the BC bus I/F, and the Host Controller, and the applications and services behind the bus I/F of the BC.

1.6.4. Requirement Numbering and Verification Methods

The Requirements in this specification are combined with Requirement Identifications.
This Requirement Identification is a combination of:

- A requirement dedicated number (for tracking purpose)

- An indication which layer of the (OSI-) communication model is involved:
  - PHL - Physical Layer
  - DLL- Data Link Layer
  - TFL - Transfer Layer

- The foreseen Verification method for this requirement:
  - N - Not to be tracked
  - T - Test
  - R - Review of Design
  - A - Analysis
  - I - Inspection
  - D - Definition

Example: 100-PHL- R, T

Meaning: Requirement no. **100** , Physical Layer, Verification method: **R**eview of Design and **T**est.

2. Physical Layer

2.1. Data Bus Topology

2005-PHL-R,T

The data bus topology shall be in accordance to AD 1, see paragraph §4.5, §30.10.

2010-PHL-T

The physical characteristics of the Bus interfaces shall be in accordance with MIL STD 1553 B, see AD1.

2015-PHL-R,I

All terminals shall use transformer coupled stubs.

2020-PHL-R,I

The BC shall use transformer coupled stubs.

2025-PHL-

Further details are TBD.

2.2. Data Bus Medium

2030-PHL-R,T

The data bus medium shall be a TSP cable with 75 Ohms TBC.

2035-PHL-R,T

See AD 1, paragraph §4.5, §30.10.

2.3. Stubs, Connectors, Shielding

2040-PHL-R,T

See AD 1, paragraph §4.5, §30.10.

2.4. Electrical Signal Characteristics

2045-PHL-R,T

See AD 1, paragraph §4.5, §30.10.

3. Data Link Layer

3.1. General Requirements

3.1.1. Bus Controller (BC) Functions

3005-DLL-R,T

The (redundant) Command and Data Management System (CDMS) shall act as Bus Controller (BC).

- 3010-DLL-R,T
- Automated message repetition by the BC in case of transmission errors is not foreseen.
-
- 3015-DLL-R
- RT to RT transmissions are not foreseen.
-
- 3020-DLL-R
- The usage of broadcast messages is allowed.
-

3025-DLL-R,T

The BC shall be able to send MIL-Bus messages to the units or instruments working as RTs on the bus.

3030-DLL-R,T

The BC shall support Mode Commands as shown in Table 3.2.4-1: List of Mode Commands, with the corresponding patterns.

3035-DLL-R,T

The BC shall support the RT SA allocation as shown in Table 3.2.3-1: SA utilization table.

3.1.2. Remote Terminal Functions

3040-DLL-R,T

All spacecraft units or instruments connected to the data bus shall act as Remote Terminals (RT).

3045-DLL-R,T

The RT shall support Mode Commands as shown in table 3.2.4-1: List of Mode Commands, with the corresponding patterns.

3050-DLL-R,T

Each RT shall support this SA allocation as shown in Table 3.2.3-1: SA utilization table.

- 3052-DLL-R
- The RT shall support broadcast messages.

3.1.3. Message Formats and Word Formats

3.1.3.1. Word Formats

3055-DLL-R,T

The used word formats shall be in accordance to AD 1. See below.

Figure 3.1.5-1: Word Formats

Bit times :

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20

Bit # (Bit Position) :

S	S	S	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	P

MIL 1553 B - Command Word:

Sync	RT Address	T/ R	Subaddress	Data Word Count	P

MIL 1553 B - RT Status Word:

Sync	RT Address	a	b	c	Reserved ´000´	d	e	f	g	h	P

a) Message Error b) Instrumentation c) Service Request d)Broadcast Command Received

e)Busy f) Subsystem Flag g) Dynamic Bus Control Acceptance h) Terminal Flag P= Parity Bit

MIL 1553 B - Data Word:

Sync	Data	P

3.1.3.2. Message Formats

3060-DLL-D

The MIL Bus transfer formats to be utilized are shown in figure 3.1.3.2-1. There are three basic types of messages, which are always initiated by a specific Command of the BC :

3065-DLL-D

- Receive Messages

These messages are sent from the BC to the addressed RT (and/or attached equipment / subsystem), which responds with dedicated actions. There is always a Status Word reply but no Data reply.

3070-DLL-D

- Transmit Messages

These messages are sent from the addressed RT or equipment / subsystems to the BC; they carry the requested data.

- 3075-DLL-D

- Broadcast Messages

These messages are used by the BC to send Commands (with or without data) to all RTs in parallel, which initiate dedicated actions at the RTs (e.g. synchronization). There is no status or data reply allowed.

Figure 3.1.3.2-1: Used Information Transfer Formats

BC to RT Transfer

RECEIVE COMMAND	DATA WORD	DATA WORD	....	DATA WORD	RESPONSE TIME	STATUS WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

RT to BC Transfer

TRANSMIT COMMAND	RESPONSE TIME	STATUS WORD	DATA WORD	DATA WORD	....	DATA WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

Mode Command without Data Word

MODE COMMAND	RESPONSE TIME	STATUS WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

Mode Command with Data Word (Transmit)

MODE COMMAND	RESPONSE TIME	STATUS WORD	DATA WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

Mode Command with Data Word (Receive)

MODE COMMAND	DATA WORD	RESPONSE TIME	STATUS WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

Broadcast BC to RT Transfer

RECEIVE COMMAND	DATA WORD	DATA WORD	....	DATA WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

Broadcast Mode Command without Data Word

MODE COMMAND	INTER-MESSAGE GAP	NEXT COMMAND WORD

Broadcast Mode Command with Data Word

MODE COMMAND	DATA WORD	INTER-MESSAGE GAP	NEXT COMMAND WORD

3.1.4. Message Timing

3080-DLL-T

The timing of messages shall be in accordance to AD 1.

Further details of Intermessage Gap and Response Time are TBD.

3085-DLL-R, T

The BC No Response Timeout shall be adjusted to TBD microsecond.

3.2. Command Word

3090-DLL-D

See figure 3.1.5-1: Word formats

3095-DLL-T

The utilization of the Command Word shall be in compliance to AD 1, § 4.3.3.5.1.

3.2.1. RT Address Field

3100-DLL-N

Each Remote Terminal (RT) connected to a MIL Bus has a unique address, which allows the Bus Controller (BC) to communicate with selected equipment. The address range for equipment is 1 to 30. Address 0 is reserved. Address 31 is used for broadcast.

3105-DLL-R, T

The RT Address shall be configurable via an external connector of the RT unit, see AD1 § 30.3.

3110-DLL-R

The RT Address of all onboard units shall be in accordance to Table 3.2.1-1: RT Address assignment.

3115-DLL-D

Table 3.2.1-1: RT Address assignment

TBD

3120-DLL-N

Notes:

- One RT Address must be reserved for BC, because of H/W and testing needs.
- An implementation example of an address connector is shown in Figure 3.2.1-1. All Address- and Parity lines have pull up resistors, so that a ‚0‘ on a line is coded by connecting it to common secondary return (secondary zero volt), as shown below. The type of Parity is odd.

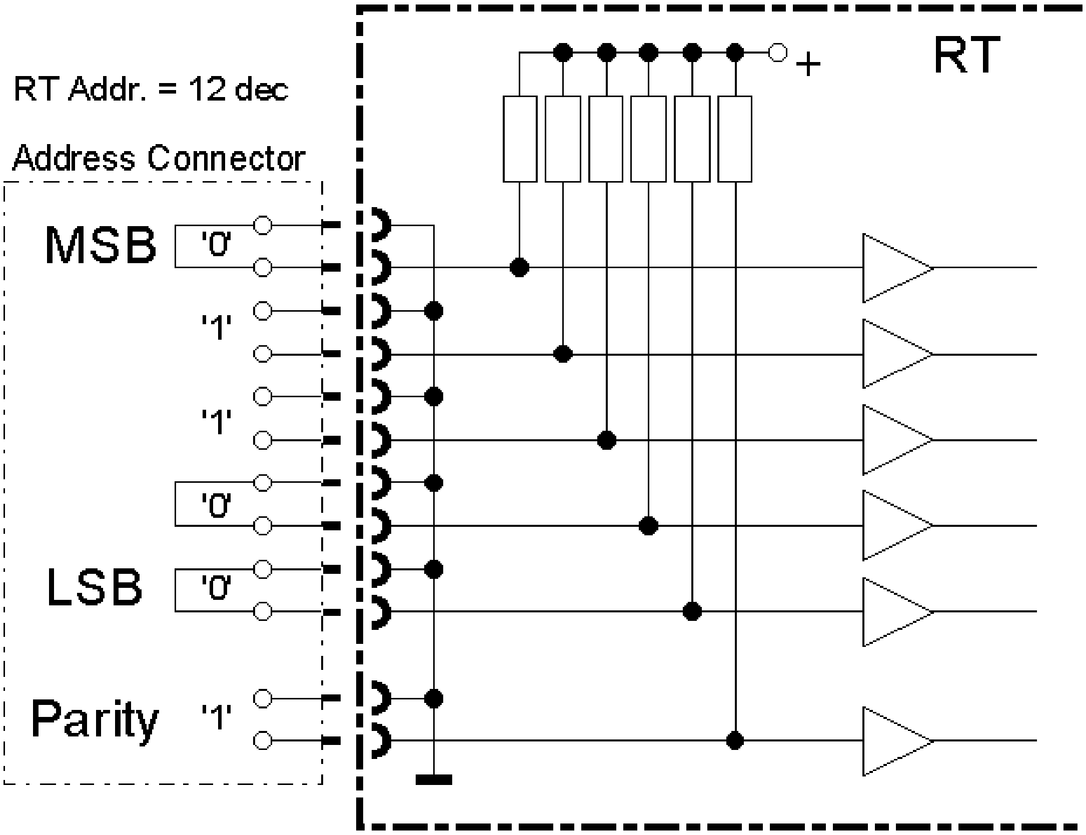


Figure 3.2.1-1 : Principle of RT Address Coding via Connector Pin Functions (Odd Parity)

3.2.2. T/R Bit Field

3125-DLL-T

The usage of this bit shall be in accordance to AD1 § 4.3.3.5.1.3.

3130-DLL-N

A logic zero indicates that the RT shall receive data from the BC and a logic one indicates that the RT shall transmit data to the BC.

3.2.3. Subaddress/ Mode Field

3135-DLL-R, T

The allocation of Subaddresses (SAs) shall be as shown in table 3.2.3-1: SA utilization table.

3140-DLL-D

Table 3.2.3-1: SA Utilization Table

Subaddress (dec.)	Transmit **T**	Receive **R**
0	reserved for Mode command	reserved for Mode command
1	Unit status data	Unit control
2	*Data send*	*Data receive*
3	*Data send*	Asynchronous Short Command 1
4	*Data send*	Asynchronous Short Command 2
5	Event Message TM - A	Event Message TM - A acknowledge
6	Event Message TM - B	Event Message TM - B acknowledge
7	*Data send*	*Data receive*
8	Internal Time TBD	Time Messages
9	*Data send*	*Data receive*
10	TM Packet Transfer Request	TM Packet Transfer Confirmation
11	TM Data send MSB	TC Data receive MSB
12	TM Data send	TC Data receive
13	TM Data send	TC Data receive
14	TM Data send	TC Data receive LSB
15	TM Data send	*Data receive*
16	TM Data send	*Data receive*
17	TM Data send	*Data receive*
18	TM Data send	*Data receive*
19	TM Data send	*Data receive*
20	TM Data send	*Data receive*
21	TM Data send	*Data receive*
22	TM Data send	*Data receive*
23	TM Data send	*Data receive*
24	TM Data send	*Data receive*
25	TM Data send	*Data receive*
26	TM Data send LSB	*Data receive*
27	TC Packet Transfer Confirmation	TC Packet Transfer Descriptor
28	Low Level Command - Status	Low Level Command

29	*Data send*	*Data receive*
30	Data Wrap read	Data Wrap write
31	reserved for Mode command	reserved for Mode command

Usage of Subadresses:

3145-DLL- T

3.2.3.1: Mode command

(SA 0, 31)

Compliant to the MIL STD 1553B, see AD1.

3150-DLL-R, T

3.2.3.2: Unit status data

(SA 1T)

Via this SA the RT shall provide H/K data and status data about the RT/ unit.

3155-DLL-T

3.2.3.3: Unit control

(SA 1R)

Via this SA the RT shall receive unit related Low Level commands.

3160-DLL- T

3.2.3.4: Asynchronous Short Command 1, 2(SA 3R, 4R)

Small Asynchronous Telecommand Packets, with a maximum length of 64 octets (one message), are send to these SAs.

3161-DLL T

The RT shall support the reception of Small Asynchronous Telecommands, as needed by the RT.

3165-DLL- T

3.2.3.5: Event Message TM - A, B

(SA 5T, 6T)

If the RT generates Event TM Packets with a maximum length of 64 octets (one message), these SAs shall be used for the exchange.

3170-DLL- T

3.2.3.6: Event Message TM - A, B acknowledge (SA 5R, 6R)

The BC shall inform the RT about the event messages acquisition by sending an acknowledge message to these SAs.

3175-DLL-R, T

3.2.3.7: Internal Time

(SA 8T)

This SA is reserved for a read of the internal time and clock status of a unit.

(Note: Details are TBD, for unit initialisation or test and troubleshooting purpose.)

3180-DLL- T

3.2.3.8: Time messages

(SA 8R)

By using this SA the BC shall distribute time information on the bus. The time message shall be send as broadcast message to all RTs.

3185-DLL- T

3.2.3.9: TM Packet Transfer Request (SA 10T)

The RT provides via this SA its request for a TM packet transfer.

3190-DLL-T

3.2.3.10: TM Packet Transfer Confirmation (SA 10R)

The BC shall place here, after reading the TM packet from the RT, the Confirmation message.

3195-DLL- T

3.2.3.11: TM Data send

(SA 11 - 26T)

This SA contains the nominal TM packet data.

3200-DLL- T

3.2.3.12: TC Data receive (SA 11 - 14R)

This SA contains the nominal TC packet data.

3205-DLL- T

3.2.3.13: TC Packet Transfer Descriptor (SA 27R)

To this SA the BC shall send the Control words to inform the RT about the presence of a new TC packet.

3210-DLL- T

3.2.3.14: TC Packet Transfer Confirmation (SA 27T)

The RT shall place here, after reading the TC packet from the BC, the Confirmation message.

3215-DLL- T

3.2.3.15: Low Level Command

(SA 28T)

This SA is reserved for low level commanding of RTs.

3220-DLL- T

3.2.3.16: Low Level Command - Status (SA 28R)

This SA is reserved for supporting a low level command execution status.

3225-DLL -R

3.2.3.17: Data send (SA 2,3,4,7,9,29T)

Unused Transmit SA.

3230-DLL-R

3.2.3.18 Data receive (SA 2,7,9,15 – 26,29R)

Unused Receive SA.

3235-DLL- T

3.2.3.19: Data Wrap read (SA 30T)

SA used for test purpose, see AD1and AD2.

The implementation of this feature is mandatory for the RT.

3240-DLL- T

3.2.3.20: Data Wrap write (SA 30R)

SA used for test purpose, see AD1 and AD2.

The implementation of this feature is mandatory for the RT.

3.2.4. Data Word Count/ Mode Code Field

3245-DLL- T

The Data Word Count Field indicates the quantity of data words to be transferred to/ from the BC. It shall be used according to AD1 § 4.3.3.5.1.5.

3250-DLL- D

In case the Data Word Count Field is set to ´00000´or ´11111´ the BC and RTs shall support Mode Commands as follows:

Table 3.2.4-1: List of Mode Commands with the corresponding patterns.

Mode Command	Pattern	Assigned	See AD 1	Supported by :			
	(bin)	SA (bin)	§ No.:	BC	S/C RTs non intelli-gent	S/C RTs	Instru-ment RTs
Dynamic Bus Control	00000	00000	4.3.3.5.1.7.1	NO	NO	NO	NO
Synchronize (without data word) *)	00001	00000	4.3.3.5.1.7.2	YES	YES	YES	YES
Transmit Status Word	00010	00000	4.3.3.5.1.7.3	YES	YES	YES	YES
Initiate Self-test **)	00011	00000	4.3.3.5.1.7.4	YES	YES	YES	YES
Transmitter (TX) Shut-Down	00100	00000	4.3.3.5.1.7.5	YES	YES	YES	YES
Override TX Shut-Down	00101	00000	4.3.3.5.1.7.6	YES	YES	YES	YES
Inhibit Terminal Flag	00110	00000	4.3.3.5.1.7.7	YES	YES	YES	YES
Override Inhibit Terminal Flag Bit	00111	00000	4.3.3.5.1.7.8	YES	YES	YES	YES
Reset Remote Terminal **)	01000	00000	4.3.3.5.1.7.9	YES	YES	YES	YES
Transmit Vector Word	10000	00000	4.3.3.5.1.7.11	YES	YES	YES	YES
Synchronize (with data word) *)	10001	00000	4.3.3.5.1.7.12	YES	YES	YES	YES
Transmit Last Command	10010	00000	4.3.3.5.1.7.13	YES	YES	YES	YES
Transmit BIT (Built-In Test data) Word **)	10011	00000	4.3.3.5.1.7.14	YES	YES	YES	YES
Selected TX Shut Down (SD)	10100	00000	4.3.3.5.1.7.15	NO	NO	NO	NO
Override Selected TX SD	10101	00000	4.3.3.5.1.7.16	NO	NO	NO	NO
Remaining Possible Command pattern (reserved for future use)	all other	00000	4.3.3.5.1.7.10 4.3.3.5.1.7.17	N/A	N/A	N/A	N/A

***) These Mode Commands are used as Broadcast Commands**

****)** According to the MIL1553B Standard, the Remote Terminal on the bus is not the total unit/equipment connected to a MIL Bus, but the interfacing circuitry only (e.g. MIL Bus chip set or H/W including the necessary buffers). E.g. the ‚Reset Remote Terminal‘ command described in the MIL 1553B Standard, does not reset the unit/equipment itself, but the interface only. The MIL Bus chip set is affected, not the unit.

3255-DLL- N

Certain functions of Mode Commands are used within the Data Bus Protocol , details are specified in chapter 4 of this specification. Further Mode Command usage is unit specific and currently TBD.

3.3. RT Status Word

3.3.1. Status Word structure

See figure 3.1.5-1: Word formats

3.3.2. Status Word bits

3260-DLL- T

After the reception of any MIL Bus message (except a MIL Bus broadcast message), a RT responds with a status word which shall be in accordance with A.D 1, § 4.3.3.5.3.

3265-DLL- D

The status word has the following content :

Bit #	Bit Meaning	Supported by RT	Status indication
0-4	Address of responding RT	YES	none
5	Message Error	YES	‚1‘=yes , ‚0‘=no
6	Instrumentation	YES	set to zero
7	Service Request	YES	‚1‘=yes , ‚0‘=no
8-10	Reserved	YES	set to zero = ´0´
11	Broadcast Command Received	YES	‚1‘=yes , ‚0‘=no
12	Busy	YES	‚1‘=yes , ‚0‘=no
13	Subsystem Flag	YES	‚1‘=fault , ‚0‘=no fault
14	Dyn. Bus Control Acceptance	YES	set to zero
15	Terminal Flag (RT fault)	YES	‚1‘=fault , ‚0‘=no fault

Table 3.3.2 -1 : MIL Bus RT Status Word

3270-DLL- T

The RT shall support the Status Word Flags as shown in Table 3.3.2 -1: MIL Bus RT Status Word.

3275-DLL- R

The RT shall support at least the Status word bit as required by AD 1 § 30.5.2., further details are TBD.

3.4. Data Word

3.4.1. Data Word Structure

3280-DLL- N

See figure 3.1.5-1: Word Formats

3.5. Data Link Layer FDIR

3.5.1. Conditions for Invalid/ Illegal Transfers on the DLL

3285-DLL-

TBD

3.5.2. Management of Bus Errors

3.5.2.1. Scope of MIL Bus-level FDIR

3300-DLL- D

The MIL Bus FDIR has the capability to manage the bus redundancy switch-over. This function collects the data necessary to monitor the status of the communications on the bus, isolates bus medium failure, and performs an automatic reconfiguration of the bus from communication medium A to communication medium B as necessary to maintain the integrity of the bus communication path.

The MIL Bus FDIR also collects FDIR data from the MIL Bus that is also relevant to RT equipment FDIR. For example, if a RT does not answer anymore on the bus, this may be because the MIL Bus communication medium is cut, or because the RT itself has failed or has been powered OFF. The bus FDIR will first try to isolate whether the error is caused by a bus failure or not. If a RT failure is confirmed instead, the failure shall be managed by a RT FDIR function. The data bus FDIR will report the error to the equipment FDIR layer.

3.5.2.2. Interfaces necessary to support Bus-level FDIR

3305-DLL- D

No specific RT interface is required to support MIL Bus FDIR. Only standard and mandatory features of the MIL-STD-1553B protocol are used.

For failure detection, the MIL Bus FDIR relies on nominal data traffic, and does not generate additional specific data traffic. Only for failure isolation a specific data traffic is generated.

3310-DLL- T

The following bus FDIR error conditions shall be collected by the MIL Bus communication service in the BC:

-	RT message error bit,	(from RT MIL-STD-1553B Status Word)
-	RT Busy bit,	(from RT MIL-STD-1553B Status Word)
-	RT subsystem flag bit,	(from RT MIL-STD-1553B Status Word)
-	RT Terminal flag bit,	(rom RT MIL-STD-1553B Status Word)
-	RT transmission error,	(from BC Bus I/F)
-	RT no response time-out,	(from BC Bus I/F)
-	BC loop back test fail,	(from BC Bus I/F)

3.5.2.3. Ground command interfaces for bus FDIR

3320-DLL- T

The following MIL Bus FDIR related commands shall be provided for usage by ground/ mission control or for usage by on-board functional layers.

- Switch to bus medium A
- Switch to bus medium B
- Disable bus automatic reconfiguration function
- Enable bus automatic reconfiguration function

3340-DLL- T

The format of these commands is TBD.

3.5.2.4. Functional Requirements of Bus-level FDIR

3345-DLL- T

By default, the BC shall use bus medium A.

3350-DLL- T

If the BC Bus FDIR isolates a failure on bus A, the whole MIL Bus traffic shall be reconfigured on bus B.

3355-DLL- T

No automatic reconfiguration from bus B to bus A shall be implemented. In case a further failure occurs on bus B, no further autonomous recovery is triggered, and the traffic is maintained on bus B.

3360-DLL- T

The BC shall provide the Bus A/B switch status to Ground/ Mission Control.

3365-DLL- T

The isolation of the bus failure shall be an automatic procedure performed by the BC, sampling the status of bus B via dedicated asynchronous messages, while the main traffic remains on bus A. A bus failure is confirmed if bus B is sampled OK, while bus A is sampled as failed, see Figure 3.5.2.4-1 below.

3370-DLL- T

If the bus channel A was used, and a bus error is detected, an automatic and global (for all messages exchanged with all RTs) switch to bus channel B is performed.

3375-DLL- T

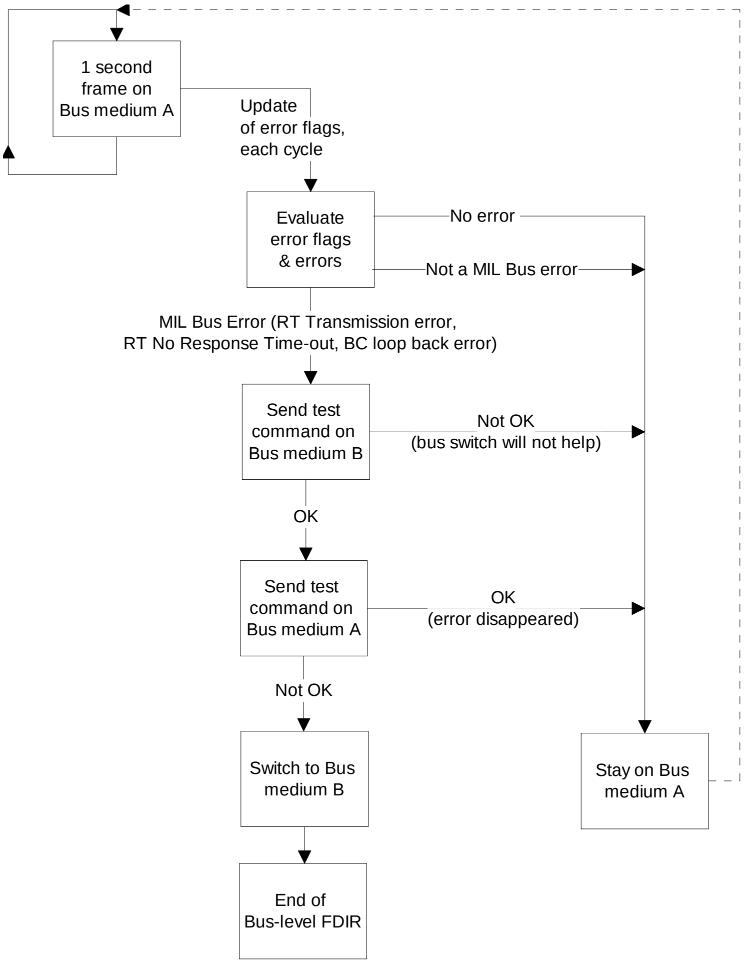
BC detected bus error condition for bus switch-over are:
- RT No response time out
- RT Transmission error
- BC Loop-back test error

Further details are TBD.

3380-DLL- T

This switch shall be effective within TBD ms.

Figure 3.5.2.4-1: Principle of MIL Bus failure isolation


1 second frame on Bus medium A
Update of error flags, each cycle
Evaluate error flags & errors
No error
Not a MIL Bus error
MIL Bus Error (RT Transmission error, RT No Response Time-out, BC loop back error)
Send test command on Bus medium B
Not OK (bus switch will not help)
OK
Send test command on Bus medium A
OK (error disappeared)
Not OK
Switch to Bus medium B
Stay on Bus medium A
End of Bus-level FDIR

3.5.2.5. Interface of Bus-level FDIR to Equipment FDIR

3385-DLL- T

The BC shall support an I/F to system-level FDIR and/ or equipment FDIR.

3390-DLL- T

The error conditions shall be reported to higher level FDIR, see Figure 3.5.2.5-1.

3395-DLL- R, T

The RTs shall provide dedicated FDIR services TBD.

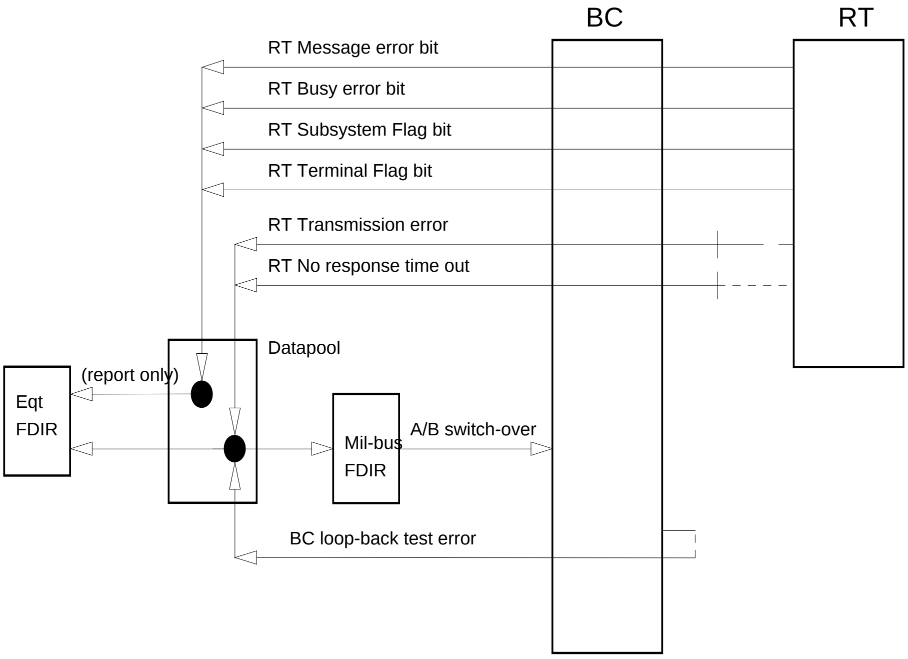


Figure 3.5.2.5-1: MIL Bus FDIR concept.

4. Transfer Layer

4.1. General Requirements

4.1.1. Functional and Performance Requirements

4005-TFL- R

- BC MIL-Bus messages and RT response messages shall be assigned to predefined time slots.
-

4010-TFL- T

- Maximum throughput for all users shall be at least 350 kbps on TM/TC packet level.
-

4015-TFL- T

- The minimum latency for a single user between the end of a packet transfer and the start of a new transfer of the same type (TM or TC) shall be 2,5 milliseconds.
-

4020-TFL- R

- The maximum size of TM-Packets shall be 1024 octets.
-

4025-TFL- R

- The maximum size of TC-Packets shall be 248 octets.
-

4030-TFL- T

- The BC and RT shall support the transfer of TM/TC-Packets with variable length.
-

4040-TFL- R

- One complete packet shall be exchanged with a user without interruptions for the user (RT).
-

4045-TFL- T

- The BC shall support each second the exchange of at least 50 TM-Packets.
-

4050-TFL- T

- The BC shall support each second the exchange of at least 16 TC-Packets.
-

4060-TFL- T

- The BC shall support each second the exchange of at least 1 time synchronization.
-

4065-TFL- T

- The BC shall support each second the exchange of at least 2 Asynchronous Short TC-Packets, with a maximum length of 64 octets, addressed independently to dedicated buffers within RTs.
-

4070-TFL- T

- The accuracy for timing and synchronization across all on-board systems up to the data interface of users shall be better than 100 microseconds TBC

4075-TFL- T

- The routing latency of an Asynchronous Short TC-Packet from the BC to the input buffer of a RT shall be below 1 millisecond.

4080-TFL- T

- The BC shall control the TM data rate of a maximum of ten users by adjusting the actual data rates according to allocated bandwidths on a 2 second time scale.

4085-TFL- T

- Each on-board user shall be served in a periodic way for at least two times per second.

4090-TFL- N

Note: The actual data traffic may be a combination of all data types described above.

4.1.2. Cyclic Transfer Protocol

4100-TFL- R, T

The cyclic Satellite Data Bus Protocol (SDBP) shall have a deterministic, periodic structure, which is synchronized with the central on-board time, implemented in the CDMS.

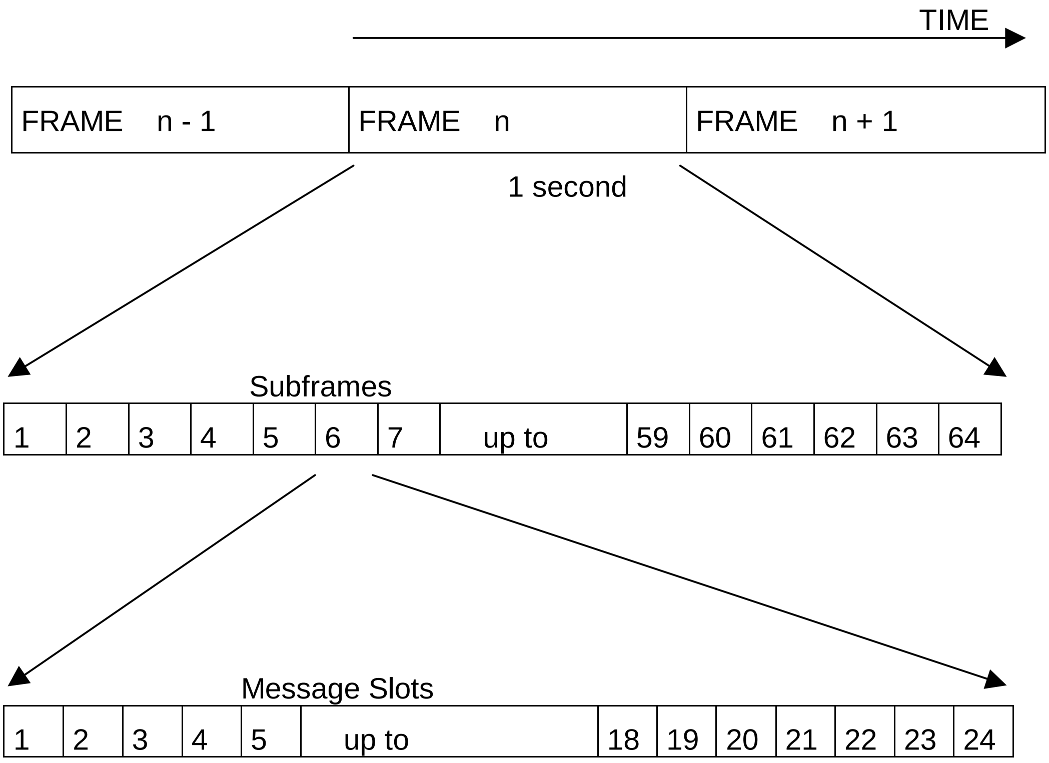
4105-TFL- R, T

The SDBP shall be based on a 1 second period called Frame. This Frame is divided into 64 Subframes, each containing a number of MIL STD 1553B messages. These messages shall occur in a Subframe within a defined timing structure called Message Slots.

4110-TFL- R, T

The TM Packet transfer from a RT (Instrument) to the CDMU shall be Subframe allocated. This means that any Instrument TM Packet shall fit into the Subframe boundaries.

Figure 4.1.2-1: Data Bus profile



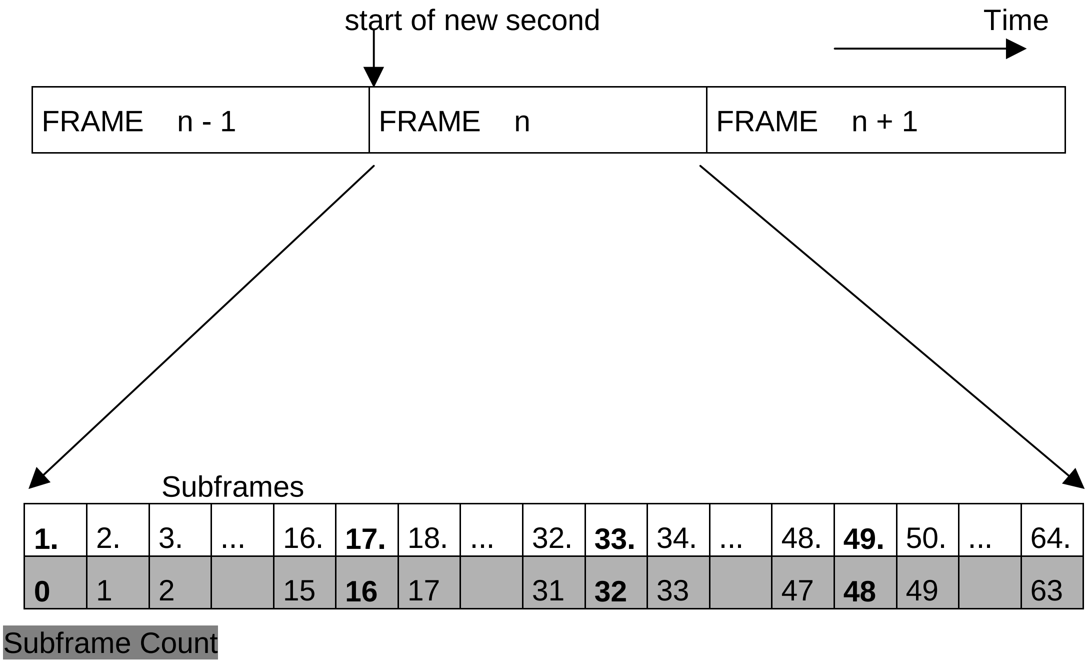
Definition: 1 FRAME = 64 SUBFRAMES ; 1 SUBFRAME = 24 MESSAGE SLOTS

Duration: 1 FRAME = 1 second: 1 SUBFRAME = 1/64 second; 1 SLOT = see table 4.1.3.1-1

4.1.3. Frame Timing

4120-TFL- R

One second / one Frame shall be divided into 64 Subframes.



1.	2.	3.	...	16.	**17.**	18.	...	32.	**33.**	34.	...	48.	**49.**	50.	...	64.
0	1	2		15	**16**	17		31	**32**	33		47	**48**	49		63

Figure 4.1.3-1: Subframe and Subframe count

4125-TFL- R, T

The first Subframe every second shall start with the Mode Command "Sync without Data word" in the first message slot.

4130-TFL- R, T

This Subframe is reserved and no instrument TM data transfer is allowed (TC packets may be sent).

4135-TFL- N

Note: This provides time for the instrument internal processes like clock (time) maintenance, etc.

Table 4.1.3-1: Subframe Utilization

Subframe	Subframe Count	Usage
1	0	First Subframe in the second, reserved for TC Packet transfer
2 to 16	1 to 15	TM Packet transfer (or TC Packet)
17	16	reserved for TC Packet transfer
18 to 32	17 to 31	TM Packet transfer (or TC Packet)
33	32	reserved for TC Packet transfer, Time information distribution
34 to 48	33 to 47	TM Packet transfer (or TC Packet)
49	48	reserved for TC Packet transfer
50 to 64	49 to 63	TM Packet transfer (or TC Packet)

4140-TFL- R, T

Subframes 1, 17, 33,49 shall be reserved for TC Packet transfer.

4145-TFL- T

In Subframe 33 the time information shall be distributed.

4150-TFL- R, T

The remaining Subframes (60 out of 64) shall be used for TM packet transfers (TC packets may be send also).

4155-TFL- R, T

The Mode Command "Sync with Data word" is inserted into the first message slot of Subframes 2 to 64.

4.1.3.1. Subframe Timing

Table 4.1.3.1-1: Message Slot Allocation Table

Slot No.	Content/ Purpose	Duration in micro seconds
1	Subframe synchronization	150
2	Command/ Acquisition Slot	750
3	Command/ Acquisition Slot	750
4	Command/ Acquisition Slot	750
5	Packet transfer	750
6	Packet transfer	750
7	Packet transfer	750
8	Packet transfer	750
9	Packet transfer	750
10	Packet transfer	750
11	Packet transfer	750
12	Packet transfer	750
13	Packet transfer	750
14	Packet transfer	750
15	Packet transfer	750
16	Packet transfer	750
17	Packet transfer	750
18	Packet transfer	750
19	Packet transfer	750
20	Packet transfer	750
21	Packet control (e.g. polling)	150
22	Packet control	150
23	Packet control	150
24	Regulation Slot	≤ 775

4160-TFL- R, T

Subframe synchronization: In the first Subframe every second the Mode Command Sync distributed as broadcast message will be used. In all other Subframes the broadcast message Mode Command Sync with Data Word will be utilized.

4165-TFL- R, T

Regulation Slot: The remaining slot time is sufficient to provide one asynchronous command insertion between any other slot without discarding the following slots. The slot time of slot No. 24 will be reduced in accordance to the asynchronous event.

4170-TFL- T

The BC shall support the Slot Allocation according to Table 4.1.3.1-1 with respect to their timing constrains.

4.1.4. Data Transfer

4.1.4.1. General

4175-TFL- N

The BC will be commanded by ground to activate one of several predefined fixed Bus Profiles. Each Bus Profile defines the function of all 64 Subframes, which belong to a cyclic 1 second frame. Each set-up is static for a longer period of time (typically for up to some hours). All instruments will be commanded independently into modes, in which they stay below their allocated maximum data rates.

4180-TFL- T

The BC shall accept ground commands for activating, deactivating, and modifying predefined Bus Profiles for data transfer.

4185-TFL- T

The BC shall support predefined Bus Profiles and shall adapt these Profiles dynamically to the RT needs.

4190-TFL- T

The RT shall accept ground commands which will start predefined operational modes.

4195-TFL- T

Each packet transfer is controlled by the exchange of a Packet Transfer Request/Descriptor and a Packet Transfer Confirmation, which are providing the necessary (handshake) information about the transfer.

4200-TFL- T

At the latest with the receiving of a next Subframe Sync Message the RT shall check the status of the packet transfer, that has taken place in the previous Subframe.

If the packet transfer was performed, then the RT

4205-TFL- T

- shall update the TM packet data buffer within 2 msec.

4210-TFL- T

- shall update the TM Packet Transfer Request Words within 2 msec.

4220-TFL- R, T

Only one TM packet transfer from each RT at a time is allowed. If there is more than one packet to be send the RT shall queue the TM packets.

4225-TFL- T

Event Messages are independent from TM transfers and their exchange shall be possible besides nominal TM packet transfers.

4230-TFL- R, T

TM packets shall be transferred within one Subframe.

4235-TFL- R, T

Transmission of multiple small TM Packets should be avoided and the RT should generate maximum length packets as much as possible for an optimised utilisation of allocated RT data rate.

4.1.4.2. Packet transfer control commands

4240-TFL- T

The BC and RT shall support Packet Transfer Requests via SA 10T and Packet Transfer Descriptors via SA 27R.

4245-TFL- D

For control purposes Packet Transfer Requests and Descriptors are introduced. Each of them consists out of two words.

Each sender, BC in case of TC packets or RT in case of TM packets, shall provide the following parameters with these words:

- The number of needed messages
- The number of words in the last message (equal to the word count pattern).

Details are defined in chapter 4.5 for TC Packet transfer and chapter 4.6 for TM Packet transfer.

4250-TFL- R, T

Each receiver shall utilize this information to re- assemble the TM or TC packet.

4.2. Frame Synchronization by BC

4260-TFL- T

In the first Subframe each second the BC shall issue the *Broadcast Mode Command Synchronize*, transmitted in the first message slot .

4265-TFL- T

The BC shall issue in all other Subframes the *Broadcast Mode Command Synchronize with Data Word*, transmitted in the first message slot .

4270-TFL- T

The distributed word formats shall be as shown below.

Figure 4.2-1: Subframe Synchronize messages

Broadcast *Mode Command Synchronize*

MIL BUS COMMAND

Sync	*RT Address* 11111	*T/R* 1	*SA* 00000	*Mode code* 00001	*Parity*

Broadcast *Mode Command Synchronize with Data Word*

MIL BUS COMMAND

Sync	*RT Address* 11111	*T/R* 0	*SA* 00000	*Mode code* 10001	*Parity*

DATA WORD

Sync	Subframe User 5 bit	Reserved bits 6 bit	Subframe Count 5 bit	*Parity*

Subframe User

4275-TFL- T

In the Subframe User field the BC shall insert the RT address of the instrument, which is allowed to send its TM data in this Subframe.

4280-TFL- T

If this Subframe is not allocated to an specific RT/ Instrument, the BC shall insert ´00000´B in this field.

4285-TFL- T

Any RT, which is not in Burst Mode, shall not interpret the field as a command or enable signal.

Subframe Count field

4290-TFL- T

In the Subframe Count field the BC shall transmit the Subframe Count value.

4295-TFL- T

The RT shall support an internal Subframe Counter and shall provide the value for BC access TBC. See chapter 4.4.

4300-TFL- T

When receiving the first Subframe each second the Subframe Counter shall be set to 0 and the RT shall increment this value by one with every received Sync with Data Word command.

4.3. Time Synchronization

4305-TFL- T

The BC shall provide system time information (Central Time Reference) via the MIL Bus.

4310-TFL- T

The time information shall be a broadcast message send to SA 8R.

4315-TFL- R

The layout of this message is shown in figure 4.3-1.

4320-TFL- T

The BC shall send the time information in Subframe no. 33 (Subframe count: 32).

4325-TFL- T

The value of the time information field shall be the time at the beginning of the next frame. Reference is the beginning of the Mode Command *Synchronize* of Subframe 1, TBC.

4330-TFL- T

The relative accuracy of the time information shall be max. 100 microseconds with respect to the system time (CTR) of the BC.

4335-TFL- R, T

The BC shall provide the Central Time Reference signal to an external test I/F for verification purpose.

4340-TFL- T

The time format shall be CUC-TAI with Coarse Time (seconds) using 4 octets and two octets for Fine Time, see RD 3.

Bit # (Bit Position) :

S	S	S	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	P

MIL 1553 B Command Word:

Sync	RT Address (= 31)	T/ R =0	Subaddress (= 8)	Data Word Count (= 4)	P

Data Words:

	MSB	LSB	
Sync	reserved set to "0000 0000"	P-Field "0010 1110"	P
Sync	T-Field : Coarse Time	T-Field : Coarse Time	P
Sync	T-Field : Coarse Time	T-Field : Coarse Time	P
Sync	T-Field : Fine Time	T-Field : Fine Time	P

Figure 4.3-1 : Time Distribution Broadcast Message

4.4. Status Polling

4350-TFL- T

The BC shall acquire the RT status data from SA 1T at least once in a second.

4355-TFL- T

The RT shall provide its health status data and additional information.

4360-TFL- T

The RT status information shall be available via SA 1T using the layout shown in figure 4.4-1 below.

4365-TFL- T

The RT shall be able to receive Low Level Commands at SA 1R. (e.g. Start of BIT, Reset, etc.)

Figure 4.4-1: RT Status Data Format.

Bit # (Bit Position) :

S	S	S	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	P

Data Words:

	MSB		LSB	
Sync	reserved	reserved	Subframe count	P
Sync	BIT data	BIT data		P
to	...	...		P
Sync	TBD	TBD		P

First data word:

4370-TFL- T

- reserved bits: These bits shall be set to zero.

4375-TFL- T

- Subframe count: Here the RT shall provide either:
  - a fixed pattern

  or
  - a copy of the bus provided Subframe count value (used for intelligent RTs)

4385-TFL- T

Details are TBD.

Second data word:

4390-TFL- R, T

In the second data word the RT shall provide:
- BIT information
- dynamic status (e.g. Watchdog)
- other health information

4395-TFL- T

For layout see table 4.4-1: RT Health Status data word.

Table 4.4-1: RT Health Status data word.

TBD(generic part, user-specific part)

4.5. Telecommand Packet Delivery

4.5.1. Nominal TC Packet Delivery Mechanism (BC to RT)

4400-TFL- T

The BC shall send in the same Subframe all messages belonging to a TC packet followed by the Packet Transfer Descriptor message.

4405-TFL- T

For TC Packet Descriptor command word layout see table 4.5.1-1.

Note: To provide bus throughput capability there are at least 4 Subframes reserved for commanding.

4410-TFL- T

The BC shall provide a circular TC Packet counter for command identification within the Transfer Layer.

4415-TFL- T

The TC packet shall be send to TC Data receive SAs, beginning with SA 11R.

4420-TFL- T

The Packet Transfer Descriptor shall be send to SA 27R.

Table 4.5.1-1: Layout of the TC Packet Transfer Descriptor (BC to RT SA 27R)

1. Data word (Packet size)				2. Data word (Packet Control)		
(3 Bit)	(5 Bit)	(3 Bit)	(5 Bit)	(6 Bit)	(2 Bit)	(8 Bit)
Reserved ´000´	No. of messages for next packet	Reserved ´000´	No. of Data Words in the last message of the next packet	Reserved, set to ´00 0000´	set to ´01´	Packet Count

4425-TFL- T

The RT shall evaluate the TC Packet Transfer Descriptor at least after receiving of the next Subframe Sync.

4430-TFL- T

The RT shall store the TC packet and copy the associated words of the Packet Transfer Descriptor to SA 27T, see table 4.5.1-2 below.

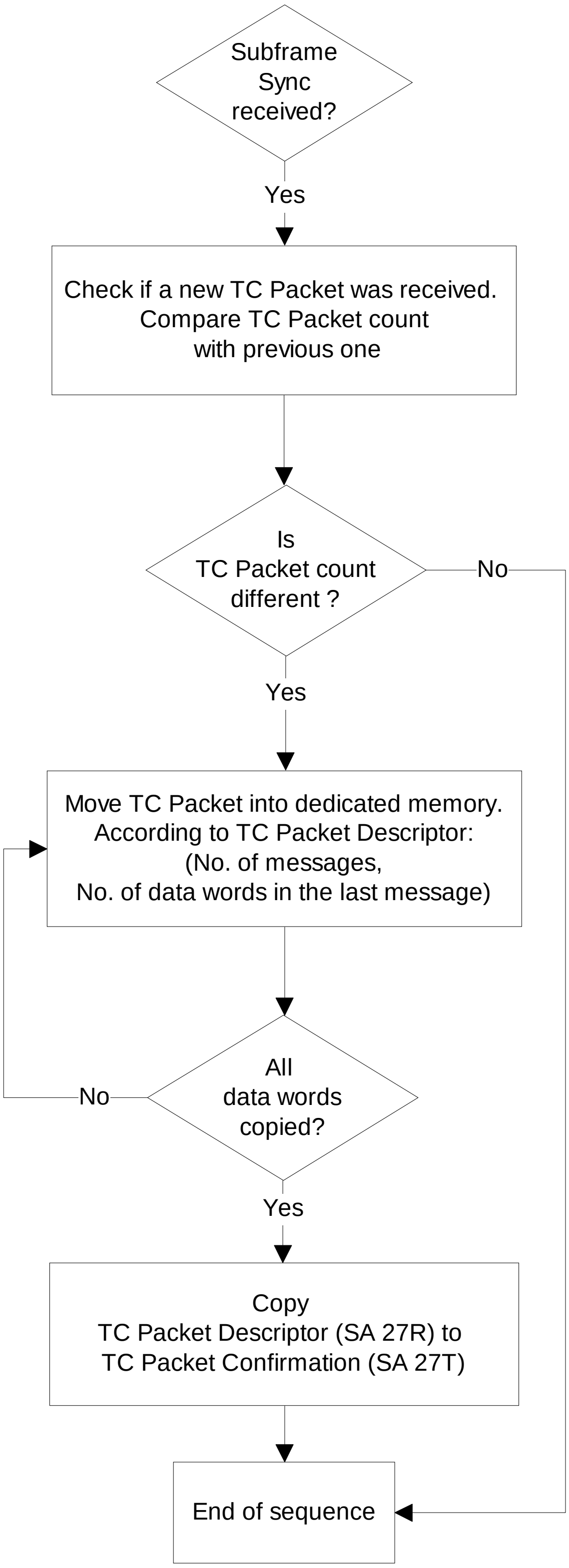
Table 4.5.1-2: Layout of the TC Packet Confirmation (from RT, SA 27T)

1. Data word (Packet size)				2. Data word (Packet Control)		
(3 Bit)	(5 Bit)	(3 Bit)	(5 Bit)	(6 Bit)	(2 Bit)	(8 Bit)
Reserved ´000´	No. of messages of the last packet	Reserved ´000´	No. of Data Words in the last message of the last packet	Reserved, ´00 0000´	Reserved ´01´	Packet Count

4435-TFL- N

For RT behavior description see figure 4.5.1-1.

Figure 4.5.1-1: Prinziple of RT behaviour for TC Packet transfer


Subframe
Sync
received?
Yes
Check if a new TC Packet was received.
Compare TC Packet count
with previous one
Is
TC Packet count
different ?
No
Yes
Move TC Packet into dedicated memory.
According to TC Packet Descriptor:
(No. of messages,
No. of data words in the last message)
No
All
data words
copied?
Yes
Copy
TC Packet Descriptor (SA 27R) to
TC Packet Confirmation (SA 27T)
End of sequence

4.5.2. Event-driven TC Packets

4440-TFL- D

Event-driven TC Packets are Asynchronous Short TC Packets *(max. 64 octets)*, which are on-board generated and serve for specific Instrument functions, which require a fast RT reaction time (e.g. On-Target Flag, Spin Reference Signal).

4445-TFL- T

The BC shall be able to send Asynchronous Short TC-Packets for special instruments control functions.

4450-TFL- T

The RT shall support the receiving of Asynchronous Short TC-Packets for special instruments control functions, as required for a certain RT.

4455-TFL- T

The routing latency of an Asynchronous Short TC-Packet from the CDMS to the input buffer of a RT shall be equal or below 1 millisecond.

4460-TFL- T

The BC shall be able to insert Asynchronous Short TC packet messages in the ongoing message sequence.

4465-TFL- T

The insertion shall not interrupt or damage any of the other bus messages in the Subframe.

4470-TFL- T

The BC shall sent event-driven TC packets of type TBD to SA 3R.

4475-TFL- T

The BC shall sent event-driven TC packets of type TBD to SA 4R.

Note: For a RT, which is required to make use of this function, a typical implementation can be, that a write-operation into SA 3R or SA 4R generates an interrupt in the bus I/F logic of the RT. This interrupt may allow the RT to determine the cause of the interrupt, and to react on the delivered TC packet as required.

4.5.3. Low level commanding

4480-TFL- T

Sending of Low Level Commands to non-intelligent RTs shall be possible via a dedicated SA (Low Level Command SA 28R).

4485-TFL- T

For low level command verification a separate Transmit SA (SA 28T) shall be supported.

4490-TFL- T

Sending of Low Level Commands to RTs for Unit Control purpose shall be possible via SA 1R.

4495-TFL- T

The BC shall support Low Level Commands according to table 4.5.1-1, Low Level Commands.

Table 4.5.3-1 Low Level Commands

Command	Bit pattern	Supported by BC	Supported by RT
TBD	TBD	TBD	TBD
TBD	TBD	TBD	TBD

Note: If the spacecraft design makes use of equipment like non-intelligent Remote Terminal Units, connected to the data bus, for housekeeping data acquisition and direct commanding, the data exchange capabilities defined in this chapter may be used for this purpose.

4.6. Telemetry Packet Retrieval

4.6.1. Periodic TM Packets

4.6.1.1. RT TM Packet Transfer Request

4500-TFL- T

The RT shall request a TM packet transfer (RT to BC) by setting its TM Packet Transfer Request control words (SA 10T).

Table 4.6.1.1-1: Layout of the Packet Transfer Request

1. Data word (Packet size)				2. Data word (Packet Control, Event handling)					
(3 Bit)	(5 Bit)	(3 Bit)	(5 Bit)	(1 Bit)	(1 Bit)	(1 Bit)	(2 Bit)	(3 Bit)	(8 Bit)
Reserved ´000´	No. of messages for next packet	Reserved ´000´	No. of Data Words in the last message of next packet	Event A	Event B	Burst Mode	Flow control	Reserved ´000´	Packet Count

- Reserved bits

4505-TFL- T

These bits are reserved for later use. The bits shall be set to zero.

- No. of messages for next packet

4510-TFL- T

This data field indicates the number of messages needed for the packet the RT is intending to send in the next Subframe. The first message of a TM Packet is always stored at SA 11T.

In accordance to table 3.2.3-1 SA utilization table.

- No. of Data Words in the last message of the next TM Packet

4515-TFL- T

This data field indicates the number of data words transmitted in the last message. In case of 32 words this field is set to "00000"B.

4520-TFL- D

Data packets always have a size of n x 16 Bit. Because they are build out of a even number of bytes there is no fill area foreseen.

- Event fields

Event A:

4525-TFL- T

0 - no Event message pending

4530-TFL- T

1 - Event message is pending BC is asked to read associated SA and to confirm.

Event B:

4535-TFL- T

0 - no Event message pending

4540-TFL- T

1 - Event message is pending BC is asked to read associated SA and to confirm.

- Burst Mode

4545-TFL- T

0 - Nominal Mode

4550-TFL- T

1 - Burst Mode

- Flow control

Meaning of Flow Control pattern:

4555-TFL- T

00 - No transfer pending

4560-TFL- T

01 - Transfer is pending

4565-TFL- T

10 - reserved

4570-TFL- T

11 - Transfer is finished (this pattern is used by BC only)

- Packet Count

4575-TFL- D

This field is used to support a RT-generated counter. By using a counter, which is incremented with every new request, and which is returned to the RT after a successful transfer, the RT is able to detect the completion of a specificTM packet transfer, even if successive packets are identical in size.

To avoid that after an RT initialisation or reset an identical packet number will be used, there is one number foreseen for that case. This number does never appear in the cyclical transmission, it is skipped in the normal sequence.

4580-TFL- T

No check on (the completeness of) the sequence will be performed by the BC depending on the Packet Count value.

4585-TFL- T

The RT shall support a circular Packet Counter (increment counter).

4590-TFL- T

The allowed range during nominal operation is 1 to 255 decimal.

4595-TFL- T

After initialization or restart the RT shall set the counter value to 0 for the first TM Packet Transfer.

4600-TFL- R, T

The RT is not allowed to use this counter for any other purpose than defined in this chapter (Packet Count).

4605-TFL- T

The BC shall identify a new TM Packet Transfer Request by comparing the Packet Counter value with the previous one.

4610-TFL- T

If these values are identical the BC shall ignore the request.

4.6.1.2. TM Packet Confirmation from the BC

4625-TFL- D

The RT, which has requested a TM packet transfer, must be able to determine, if the packet transfer was performed and the next packet data can be loaded to the message buffers. After a successful packet transfer the BC sends a handshake signal to the sending RT (TM Packet Confirmation). This handshake informs the RT which packet was the last one transmitted, by returning a modification of the Packet Transfer Request including the last Packet Count value and a transfer status.

4630-TFL- T

The BC shall support a TM packet transfer handshake by using the TM Packet Transfer Confirmation.

4635-TFL- T

The BC shall send this Command to SA 10R.

4640-TFL- T

The layout is shown in the table 4.6.1.2-1, see below.

4645-TFL- T

The BC shall send the TM Packet Confirmation in the same Subframe in which the TM Packet transfer was performed.

Table 4.6.1.2-1: Layout of the TM Packet Transfer Confirmation

1. Data word (Packet size)				2. Data word (Packet Control)					
(3 Bit)	(5 Bit)	(3 Bit)	(5 Bit)	(1 Bit)	(1 Bit)	(1 Bit)	(2 Bit)	(3 Bit)	(8 Bit)
Re-served ´000´	No. of messages of the last TM packet	Re-served ´000´	No. of Data Words in the last message of the last packet	Re-served ´0´	Re-served ´0´	Burst Mode	Flow control	Re-served ´000´	Packet Count

4650-TFL- D

The first data word is a duplication of the first Packet Transfer Request control word acquired from the RT:

The second data word shall contain all necessary information for the handshake purpose:

- Packet Count

4655-TFL- T

This field shall be identical to the packet count field of the TM packet which was acquired.last.

4660-TFL- R

The packet count value itself is with no further meaning.

- Flow control

4665-TFL- T

Flow Control is set by the BC to:11 - Meaning of Flow Control pattern: Transfer is finished.

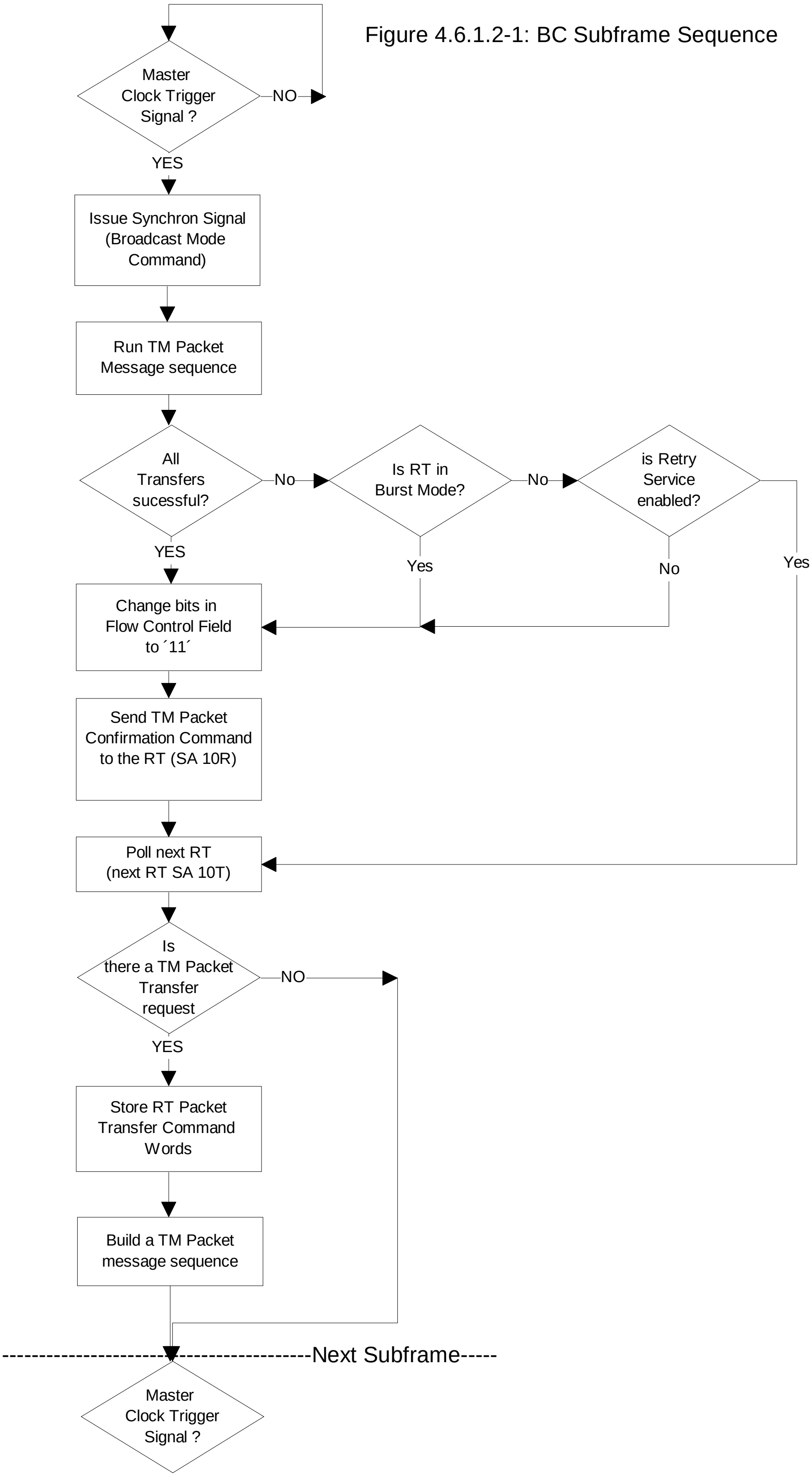
4670-TFL- D

Note: No handling of Event packets is performed via this data word. All other bits are set to zero.

4675-TFL- N

Figure 4.6.1.2-1 illustrates the logical flow of BC Sequence behavior.

Figure 4.6.1.2-1: BC Subframe Sequence


Master Clock Trigger Signal ?
NO
YES
Issue Synchron Signal (Broadcast Mode Command)
Run TM Packet Message sequence
All Transfers sucessful?
No
Is RT in Burst Mode?
No
is Retry Service enabled?
YES
Yes
No
Yes
Change bits in Flow Control Field to ´11´
Send TM Packet Confirmation Command to the RT (SA 10R)
Poll next RT (next RT SA 10T)
Is there a TM Packet Transfer request
NO
YES
Store RT Packet Transfer Command Words
Build a TM Packet message sequence
--Next Subframe-----
Master Clock Trigger Signal ?

4.6.1.3. TM Packet Transfer mechanism

4685-TFL- T

- Initialization

After the BC has been commanded by ground to execute a predefined bus traffic profile (or the BC makes use of a default profile after initialisation/ reset) it shall starts the polling of the RTs.

The RT shall request a TM packet transfer (RT to BC) by setting its TM Packet transfer control words (SA 10 Transmit).

At least one Subframe before the next scheduled TM packet transfer for a certain RT, the BC shall poll the RT for a need of transfer. This will be done by reading the RT TM Packet Transfer Request words from SA 10T.

4690-TFL- T

- Packet Delivery

If there was a packet transfer request, the BC shall acquire this TM Packet within the next Subframes by using as much transmit message commands as requested by the RT.

After the valid transmission (no error occurred) the BC sends the Transfer Confirmation message to the RT, see chapter 4.6.1.2 .

The RT shall check the content of the Confirmation message at the latest after receiving of the next *Mode Command Synchronize* .If the transfer is confirmed by the BC, the RT shall prepare the next data packet transfer.

If a new TM packet shall be sent, the RT shall load the new TM Packet Data and the new TM Packet Request Command within 2 m sec after beginning of the Subframe.

Figure 4.6.1.3-1 illustrates the logical flow of RT Sequence behavior.

4695-TFL- T

- End of Transfer

In case there is no new TM packet pending the RT shall set the first word of the TM Packet Transfer Request to ´0000 0000´B, and the Packet Count value of the second word shall stay unchanged.

The Flow Control field bits shall be set to ´00´.

4700-TFL- T

- Burst Mode

Burst Mode is used in the case that the needed data throughput for one RT is so high that several consecutive Subframes have to be used to acquire the TM Packet data from this RT.

The RT in Burst Mode shall use fixed TM Packet sizes.

The TM Packets shall have their maximum size of 1024 octets.

The Burst Mode shall be indicated by the RT by setting the bit in the Burst Mode Field of the TM Packet Transfer Request Words (see chapter 4.6).

After the BC has acquired the RT´s TM Packet Transfer Request message, the BC shall acquire in one of the following Subframes the TM Packet (SA 11T to SA 26T), and shall send TM Packet Transfer Confirmation Command words belonging to this TM Packet to SA 10R.

The BC shall perform the polling of the RT in burst mode by reading the RT TM Packet Transfer Request Words (SA 10T) immediately after the acquisition and confirmation of the TM Packet.

The BC shall not check the validity of a last packet transfer and shall not support any retry on packet level.

In the Burst Mode the RT shall provide an update of the TM Packet Transfer Request (SA 10T) and the TM Packet Transfer messages (SA 11T to SA 26T) according to following timing:

In the Burst Mode the TM Packet Transfer Request SA is updated either after:

- The Subframe has started and the RT address of the RT in Burst Mode was transmitted in the Data word of the Synchronization message TBC.

or after

- The requested TM Packet Transfer has actually started in this Subframe. This is detected by the BC-access to SA 11T.

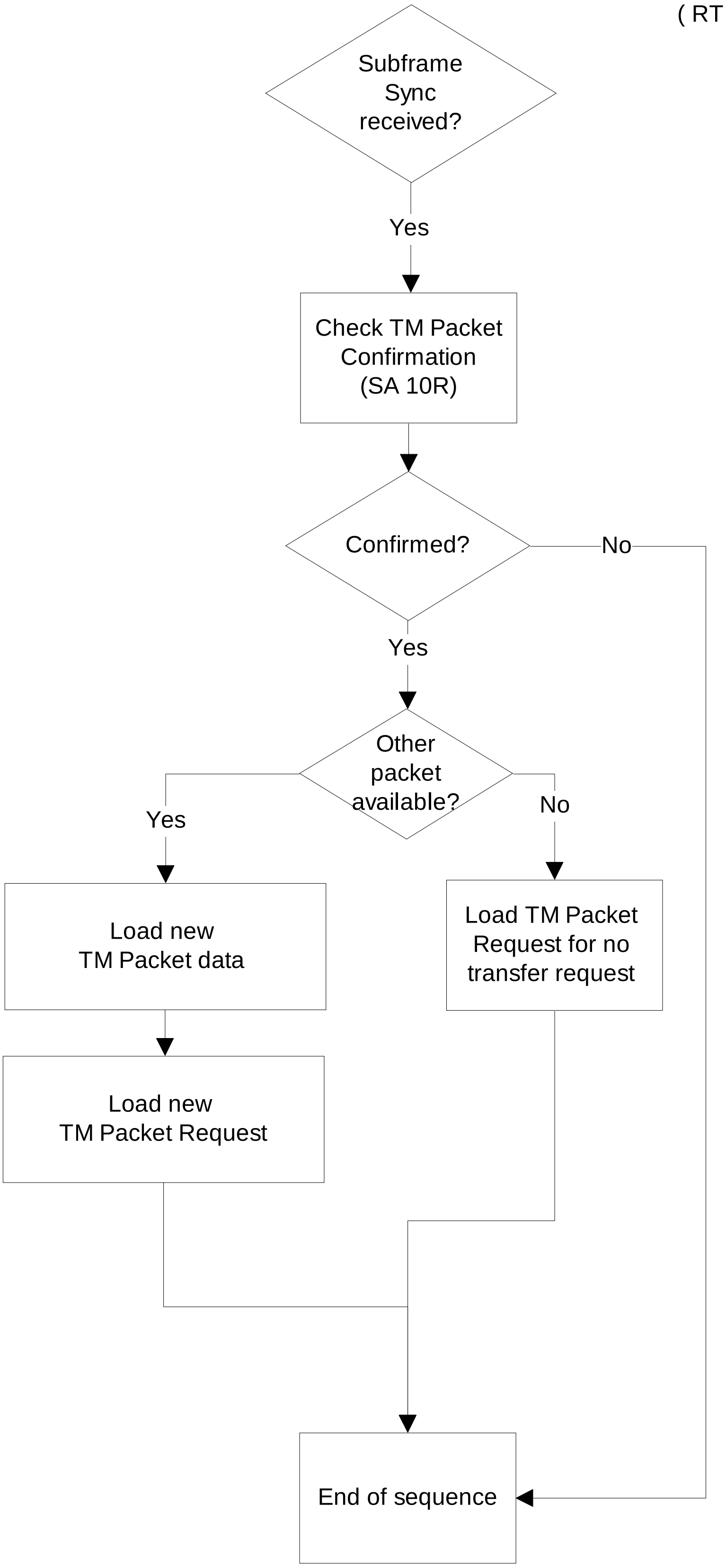
In the Burst Mode the TM Packet output buffers (SA 11T to SA 26T) shall be updated either after

- Receiving of the BC TM Packet Transfer Confirmation message (SA 10R)

or after

- The next Subframe is started and the RT address of the RT in Burst Mode was transmitted within the Synchronization message TBC.

Figure 4.6.1.3-1: RT TM Packet flow diagram
(RT is not in Burst Mode)


Subframe Sync received?
Yes
Check TM Packet Confirmation (SA 10R)
Confirmed?
No
Yes
Other packet available?
Yes
No
Load new TM Packet data
Load TM Packet Request for no transfer request
Load new TM Packet Request
End of sequence

4.6.2. Event-driven TM Packets

4800-TFL- D

TM-Event packets are small TM data packets, which are used for Telecommand verification and for reporting asynchronous RT events; details are defined in the main part of RD 2. They fit into one MIL Bus message. These TM Event messages are handled independently from the nominal TM packet transfer.

4805-TFL- T

If the RT has to start a TM Event transfer it shall load the Event packet message into the buffers associated with Event message SA 5T or SA 6T (see Table 3.1-2) and shall set the corresponding Event Flag in the Packet Transfer Request control words.

4810-TFL- T

The Event transfer handshake is performed by a duplication of the Event message coming from the RT Transmit SA to the corresponding Receive SA by the BC.

4815-TFL- T

By comparing the contents of these SAs, the RT shall be able to detect that the BC has acquired a certain Event message.

4820-TFL- T

The BC shall take care that an Event Flag, which is taken into account, will initialize only once the associated reading of an Event message.

4830-TFL- T

After the BC has started the transfer of an Event message in a certain Subframe, it shall ignore status changes of the associated Event Flag during this Subframe and the next Subframe.

Note: This provides time for the RT for resetting of this flag.

4835-TFL- R, T

The RT shall check at least after every Subframe sync if the Event Packet Transfer is confirmed.

4840-TFL- T

If the Event Packet Transfer is confirmed the RT shall reset the corresponding Event Flag.

4845-TFL- R, T

A new Event Flag setting of the same kind (A or B) is only allowed after waiting 2 Subframes.

4850-TFL- R, T

The RT shall use Event TM-A messages (SA 5T) for TBD.

4855-TFL- R, T

The RT shall use Event TM-B messages (SA 6T)for TBD.

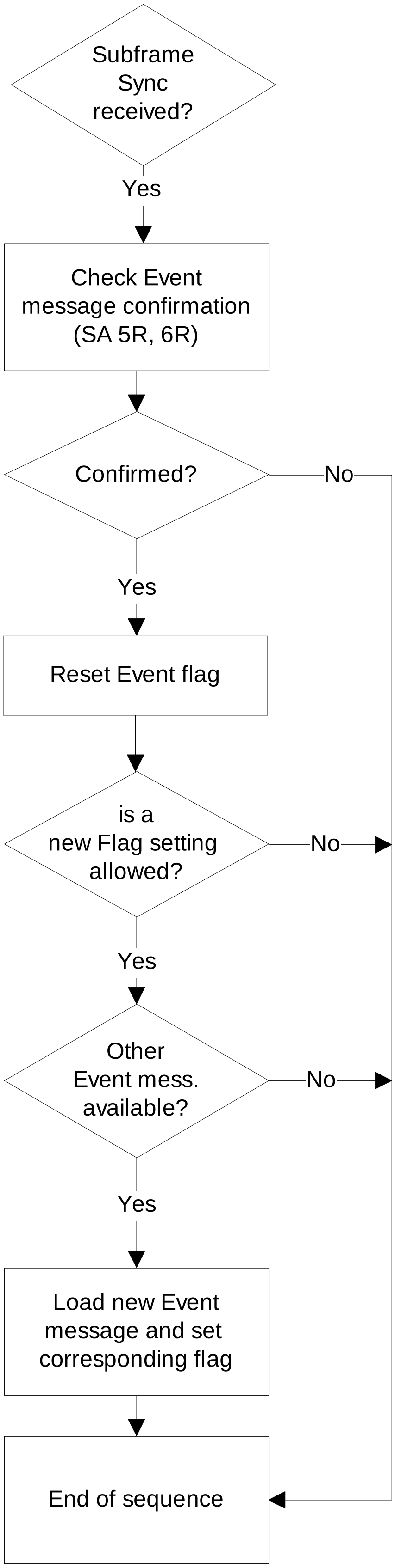
4860-TFL- T

The BC shall poll a RT at least two times a second to detect an Event Flag setting.

4865-TFL- N

Figure 4.6.2-1 illustrates the logical flow of the RT Event handling behavior.

Figure 4.6.2-1 : RT Event handling


Subframe Sync received?
Yes
Check Event message confirmation (SA 5R, 6R)
Confirmed?
No
Yes
Reset Event flag
is a new Flag setting allowed?
No
Yes
Other Event mess. available?
No
Yes
Load new Event message and set corresponding flag
End of sequence

4.7. Transfer Layer FDIR

4900-TFL-

TBD

4.8. Data Rate Monitoring

4910-TFL- T

The BC bus protocol layers shall provide information to the higher application layers of the BC about the amount of acquired TM data per RT and per second.

4915-TFL- N

Further details are TBD.

End of Document