European Space Agency

Directorate of Scientific Programme / Scientific Projects Department
Directorate of Technical and Operational Support / Mission Operations Dept.

FIRST / PLANCK

Packet Structure
Interface Control Document

SCI-PT-ICD-07527

Issue Draft 1.0
22 May 2000

Approved by: ______________________
F. Vandenbussche (ESTEC)

Approved by: ______________________
J. Dodsworth (ESOC)

Prepared by: ______________________
S. Thürey (SCI-PXI),

CHANGE RECORD SHEET

Date	Issue No.	Rev. No.	Pages Affected	Description/Authority	CR No.
22-May-2000	Draft 1.0		All	Draft issue, updated after review by ESA and PIs	---

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, Milano	1

Recipient	**Organisation**	**Number of Copies**

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 ECSS 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 a mission-specific way. Therefore, this document specifies in an **appendix** a Satellite Data Bus Protocol, which provides the necessary definitions for controlling the on-board data transfer within the **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 spacecrafts and their payloads will be conducted under control of the Mission Operations Centre (MOC) 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 together with their data structures. 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, Draft 5.0

1.4 REFERENCE DOCUMENTS

1. Packet Utilisation Standard (PUS), ECSS-E-70/41, Draft 04, April 1999
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 1, September 1978

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	
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.

PACKET HEADER (48 bits)							**PACKET DATA FIELD (VARIABLE)**		
PACKET ID				PACKET SEQUENCE CONTROL		PACKET LENGTH	DATA FIELD HEADER	APPLIC-ATION DATA	PACKET ERROR CONTROL
Version Number	Type	Data Field Header Flag	APID	Sequence Flags	Sequence Count				
3	1	1	11	2	14				
16 bits				16 bits		16 bits	32 bits	N x 16 bits	16 bits

Figure 3.1-1 Telecommand Packet Fields

3.1.1 Packet Header

3.1.1.1 Packet ID

Version Number:
The Version Number must be set to '000'$_{BIN}$ for all commands.

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

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.

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

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

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

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	Segment-ation 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

Version Number:
The Version Number must be set to ’000’$_{BIN}$ (The specification in this document is consistent with [RD-4] 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.

Type:
For telemetry source packets, the type must be set to zero.

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).

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

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.

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

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 onboard 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.

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.

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.

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 Sub-type 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 $^{\text{Duration}}$

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 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	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.

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 using Absolute Addresses Report (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 using Absolute Addresses Report (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 5]**, 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 should 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 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

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. 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 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 5]**, 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 Activate Function Telecommand. 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 Activate command, 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:

Activate Function (8,1)

Telecommand Packet, Application Data:

Function ID	N	Parameters
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 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
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
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.

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 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 Parameters 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 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) to this Application, and then sending a synchronisation signal or message to one or more end-users and freezing a copy of the on-board master time **at this moment**. As the addressed end-user or Application has to start its local clock from Zero at the reception of the synchronisation signal, the sum of this local time and the copy of the master time, which can be delivered and added at any time later, gives an exact copy of the on-board master time.

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'$_{BIN}$.

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 the next synchronisation signal or message and starts its local clock from Zero.

Application Data : None

5.9.1.3 TC (9,5) : Add Time Code

On receipt of this command, the user **adds** the time information delivered in the data field to the local representation of the on-board time, in order to generate a copy of the on-board master time. He may immediately reload his local clock with this value.

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.

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 conceptare 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:

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

N2:

The number of Application IDs that follow

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.

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 Telecommand 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.

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:

Whether the release of this Telecommand is dependent on the success or failure of the Telecommand with which it is interlocked, viz.:

"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.

The Scheduling Event may take other mission-specific values, TDC.

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 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.

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).

5.11.1.4.2 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	

N2	Application Process ID
Unsigned Integer	Enumerated
	Repeated N2 times

Range:

The value of Range is 1.

N2:

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.

5.11.1.6.1 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.

5.11.1.6.2 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

5.11.1.6.3 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

5.12.1 Telecommand Packet Data Field Structure:

TBD

5.12.2 Telemetry Packet Data Field Structure:

TBD

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

5.15.1 Telecommand Packet Data Field Structure:

TBD

5.15.2 Telemetry Packet Data Field Structure:

TBD

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

5.18.1 Telecommand Packet Data Field Structure:

TBD

5.18.2 Telemetry Packet Data Field Structure:

TBD

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 :

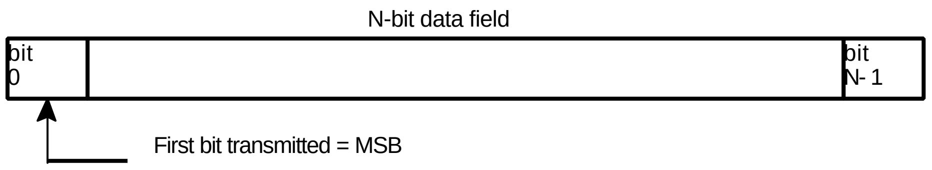

Figure A1.1 Bit numbering

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 AND TC PACKET TYPES AND SUBTYPES for FIRST / Planck

To be written

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		
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:

$L(X) = S^{15}_{i=0} X^i$ (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
APID	Application Process Identifier
CDMS	Command and Data Management System, On-board Data Handling System
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
HK	Housekeeping
HL	High Limit
ICD	Interface Control Document

Acronym	Description
ID	Identifier
ISO	International Standards Organisation
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
PB	Play Back
PEC	Packet Error Control
PID	Process 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

Acronym	Description
SCET	Spacecraft Elapsed Time
SCOE	Special Check Out Equipment
SDU	Service Data Unit
SID	Structure Identifier
SOC	Science Operations Centre
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).
(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.
FID	Function Identifier, identifies a function of a task and defines the structure of the parameter field in the packet. The same FID may be used by different APIDs.
MID	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.
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.
Parameter ID	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.
SID	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.
TID	Task identifier, identifies a task within an application. The same TID may be used by different APIDs.
Task	A definite amount of actions to bring about a result for a user. One or more tasks may be active simultaneously within a application process.
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	Segment-ation 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} [1].

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.

[1] The specification in this document is consistent with [RD-2] and supersedes [AD-1] with respect to the Version Number.

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

Spacecraft Time Source 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	Segment-ation 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} [2].

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.

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.

[2] The specification in this document is consistent with [RD-4] and supersedes [AD-1] with respect to the Version Number.

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 freely chosen 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 Satelite Data Bus Protocol

To be written

End of Document