Telecommand and Telemetry Formats
Software Maintenance Focus (detailed)

Large Area Telescope (LAT) Flight Software (FSW) Group
Gamma Ray Large Area Space Telescope (GLAST) Project

Stanford Linear Accelerator Center (SLAC)
Menlo Park, CA, USA

(generated on May 3, 2005)

Table of Contents

0 SCOPE

0.0 Identification

This Interface Control Document (ICD) describes the formats and protocols associated with telecommands and telemetry for the Gamma-ray Large Area Space Telescope's (GLAST) Large Area Telescope (LAT) payload.

0.1 System Overview

GLAST is a high-energy gamma-ray observatory, designed for making observations of celestial sources in the energy band extending from 20 MeV to 300 GeV, with complementary coverage between 10 KeV and 25 MeV for gamma-ray bursts.

The LAT instrument detects both Cosmic Rays (i.e., charged particles) and Gamma Rays (i.e., high-energy photons), capturing the resulting information as "events". The LAT Flight Software (FSW) is tasked with configuring and operating the instrument, as well as deciding which events are Gamma Rays from celestial sources.

The vast majority of communication with the FSW is accomplished by the exchange of telecommand and telemetry packets. Telecommand packets are used to load new software, set configuration values, initiate operations, etc. Telemetry packets are used to report data of various types, including diagnostic, housekeeping, and science.

The FSW receives telecommand packets from the spacecraft (SC) and telemetry packets from the Gamma-ray Burst Monitor (GBM). It sends telemetry packets to the spacecraft, for retransmission to Earth-based installations. On rare occasions, the FSW sends telecommand packets (e.g., repoint requests) to the spacecraft.

0.2 Document Overview

This document details the telecommand and telemetry interfaces between the LAT payload and the GLAST spacecraft bus. It is organized as follows:

0 Scope

Discussion of the general nature of the document.

1 References

Documents, acronyms, and glossary terms referenced in or required for use with this document.

2 Interface Overview

A summary of the telecommand and telemetry interfaces between the LAT payload and the GLAST spacecraft bus and between the LAT payload and the GBM payload.

3 CCSDS Protocol

A summary of the CCSDS protocol.

4 Package Overview

A summary of the packages described in subsequent sections.

5 <package> (e.g., FILE, LCAT, LHK):

Definitions of relevant items (e.g., packets, attributes) for the package in question.

Indexes

Indexes into the package chapters, organized by APID, item name, etc.

0.3 Document Data Sources

The "front matter" for this document was derived, largely, from a set of hand-edited configuration files. Consequently, it may not track all changes in the software, etc.

The telecommand and telemetry descriptions in this document were autogenerated from the following packages/versions in the LAT flight software code management system:

0.4 Document Output Variants

The "Telecommand and Telemetry Formats" document is available in four variant formats, differing in focus ("Software Maintenance", "Instrument Operation") and level of detail ("abridged", detailed").

The Software Maintenance variants are aimed at the needs of software maintainers. Consequently, they discuss data structures (e.g., Bitfields), use LCAT (i.e., C) nomenclature, etc.

The Instrument Operation variants are aimed at the needs of instrument operators. Consequently, they ignore data structures, use ITOS nomenclature, etc.

None of these variants is as complete and navigable, however, as the online (web-based) variants. If online access is available, this should be your first choice.

1 REFERENCES

This section lists documents, acronyms, and glossary terms that either are referenced in this Interface Control Document or provide additional information applicable to the understanding of this document.

1.0 Applicable Documents

LAT Project Documents

LAT Project Documents	
Document Number	**Document Title**

GLAST Project Documents

GLAST Project Documents	
Document Number	**Document Title**

NASA Standards and Guidelines

NASA Standards and Guidelines	
Document Number	**Document Title**
CCSDS 101.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Telemetry Channel Coding, May 1999
CCSDS 102.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Packet Telemetry, November 1995
CCSDS 200.0-G-6	Consultative Committee for Space Data Systems (CCSDS) Report for Telecommand: Summary of Concept and Rationale, January 1987
CCSDS 201.0-B-3	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Telecommand: Part 1, Channel Service, June 2000
CCSDS 202.0-B-2	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Telecommand: Part 2.1, Command Operation Procedures, October 1991
CCSDS 203.0-B-1	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Telecommand: Part 3, Data Management Service Architectural Specification, January 1987
CCSDS 102.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Packet Telemetry, November 1995
CCSDS 102.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Packet Telemetry, November 1995
CCSDS 102.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Packet Telemetry, November 1995
CCSDS 102.0-B-4	Consultative Committee for Space Data Systems (CCSDS) Recommendation for Packet Telemetry, November 1995

Military Standards and Guidelines

Military Standards and Guidelines	
Document Number	**Document Title**

Commercial Standards

Commercial Standards	
Document Number	**Document Title**
RFC-1590	Media Type Registration Procedure
RFC-1591	Domain Name System Structure and Delegation

1.1 Acronyms

Acronyms	
Acronym	**Definition**

1.2 Glossary

Glossary	
Term	**Definition**

2 INTERFACE OVERVIEW

This section provides a summary description of the telecommand and telemetry interfaces between the LAT payload and the GLAST spacecraft bus and between the LAT payload and the GBM payload.

2.0 Data Representation Conventions

Unless otherwise specified, the following data representation conventions are applicable for the entire document:

- Bits are numbered from 0 to N, where 0 represents the least significant bit of the field and N represents the most significant bit in a field. TBR - is this appropriate for the RAD750 and spacecraft bus? (BD)
- The data representations in this document treat bytes as the smallest addressable unit size.
- When multiple bytes are combined to form larger data units, the most significant byte of the field is the byte with the lowest address.
- When data are transferred across a serial interface, the bits flow from the most significant bit to the least significant bit.
- Floating-point values are represented in either a 32-bit IEEE-754 format or a 64-bit IEEE-754 format.

Refer to Figure 1-1 and Figure 1-2 for graphical depictions of the data representation conventions.

TBR - depending on the answer above, these pictures may need to change to reflect the convention for bit number assignments.

>>> fig_01.gif

>>> fig_02.gif

Time is represented as a 64-bit value from a time epoch. The time epoch for LAT and the spacecraft bus is 00:00:00.0 hours of January 1st, 2001. That is, the midnight between December 31st, 2000 and January 1st, 2001. The 64-bit value is represented in Figure 3. The 32-bit Timestamp Seconds represents the number of elapsed seconds since the epoch. The Timestamp Sub-Seconds represents the number of micro-seconds elapsed since the last second.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Timestamp Seconds MSW															
Timestamp Seconds LSW															
Timestamp Sub-Seconds MSW															
Timestamp Sub-Seconds LSW															

2.1 Interface Identification

The LAT interfaces are depicted below.

>>> fig_04.gif

Telecommands to LAT

Interface:	1553
Type:	BC to RT
Source:	Bus Controller - Spacecraft Bus Processor
Destination:	Remote Terminal 3 - LAT SIU
Data Transfer:	Subaddress 27 - 62 Bytes
Transfer Rate:	20 Hz maximum
Transfer Gap:	20 milliseconds minimum

Protocol: BC to RT data transfer will only take place when a LAT telecommand is ready for transfer.
Message Type: CCSDS Version 1 Telecommand Packet
Message Size: 62 bytes maximum, even number of bytes, zero filled to 62 bytes
APID Range: 0x640 - 0x69F, directed to LAT
0x701, broadcast from spacecraft bus processor
0x703, broadcast from GBM
Notes: None

Telecommands from LAT

Interface: 1553
Type: RT to BC
Source: Remote Terminal 3 - LAT SIU
Destination: Bus Controller - Spacecraft Bus Processor
Data Transfer: Subaddress 29 - 64 Bytes
Transfer Rate: 5 Hz
Transfer Gap: 150 milliseconds minimum to 250 milliseconds maximum
Protocol: First word of the transfer is a Transfer Request Counter. If the counter has changed since last read by the BC and is non-zero, then the transfer contains a new telecommand from the LAT.
Message Type: CCSDS Version 1 Telecommand Packet
Message Size: 62 bytes maximum, even number of bytes, zero filled to 62 bytes
APID Range: 0x600 - 0x63F, directed to spacecraft bus processor
0x6A0 - 0x6FF, directed to GBM
0x702, broadcast from LAT
Notes: The GBM telecommands and LAT broadcast telecommands are routed to the GBM by the spacecraft bus processor.

Housekeeping Telemetry from LAT

Interface: 1553
Type: RT to BC
Source: Remote Terminal 3 - LAT SIU
Destination: Bus Controller - Spacecraft Bus Processor
Data Transfer: Subaddress 11 through Subaddress 25 - 960 Bytes
Transfer Rate: 4 Hz
Transfer Gap: 500 milliseconds maximum (question - what is the minimum?)
Protocol: First word of the transfer is a Transfer Request Counter. If the counter has changed since last read by the BC and is non-zero, then the transfer contains new telemetry from the LAT. If the BC successfully reads the 15 subaddresses, then a BC to RT transaction is performed to write one data word to RT-3 subaddress 26. The fifteen BC to RT transactions and the sole RT to BC transactions are all scheduled to occur consecutively. By LAT convention, the housekeeping telemetry will occupy the first 116 bytes available for telemetry transfer (last 62 bytes of SA 11, first 54 bytes of SA 12). Multiple telemetry packets can be transferred during one transaction. The packets must be contiguous. Packets cannot span transactions. If the packet data for a transaction uses 958 or less bytes, then the last packet must be followed by at least two bytes containing zeroes.
Message Type: CCSDS Advanced Orbital Systems Telemetry Packets
Message Size: 116 bytes
APID Range: 0x200 - 0x25F from LAT

Notes: LAT Housekeeping telemetry is transmitted when a TDRSS or GN RF link is active. The spacecraft bus processor stores the housekeeping telemetry.

Diagnostic Telemetry from LAT

Interface:	1553
Type:	RT to BC
Source:	Remote Terminal 3 - LAT SIU
Destination:	Bus Controller - Spacecraft Bus Processor
Data Transfer:	Subaddress 11 through Subaddress 25 - 960 Bytes
Transfer Rate:	4 Hz
Transfer Gap:	500 milliseconds maximum
Protocol:	Reference protocol for Housekeeping Telemetry from LAT.
Message Type:	CCSDS Advanced Orbital Systems Telemetry Packets
Message Size:	942 bytes maximum, even number of bytes
APID Range:	0x260 - 0x33F from LAT
Notes:	LAT Diagnostic telemetry is transmitted when a TDRSS or GN RF link is active. The spacecraft bus processor stores the diagnostic telemetry. (question - how does the spacecraft processor rate-control the diagnostic telemetry?)

Alert Telemetry from LAT

Interface:	1553
Type:	RT to BC
Source:	Remote Terminal 3 - LAT SIU
Destination:	Bus Controller - Spacecraft Bus Processor
Data Transfer:	Subaddress 11 through Subaddress 25 - 960 Bytes
Transfer Rate:	4 Hz
Transfer Gap:	500 milliseconds maximum
Protocol:	Reference protocol for Housekeeping Telemetry from LAT.
Message Type:	CCSDS Advanced Orbital Systems Telemetry Packets
Message Size:	942 bytes maximum, even number of bytes
APID Range:	0x340 - 0x39F from LAT
Notes:	LAT Alert telemetry is transmitted when a TDRSS or GN RF link is active. If no RF link is active, then the spacecraft bus will activate the TDRSS link to transmit an alert telemetry packet. The spacecraft bus processor stores the alert telemetry.

Science Telemetry from LAT

Interface:	LVDS
Type:	LAT Protocol (LATp)
Source:	LAT - GASU
Destination:	Spacecraft Bus - Solid State Recorder
Data Transfer:	Contiguous and Encapsulated 128-bit LATp cells. Encapsulation uses a cell announce, cell truncate, and cell parity for each cell. The first cell in contiguous sequence has a cell header.
Transfer Rate:	TBD bits/second
Transfer Gap:	TBD time between LATp transfers
Protocol:	TBD protocol Packets must be a multiple of bytes in length.
Message Type:	CCSDS Advanced Orbital Systems Telemetry Packets
Message Size:	65540 bytes maximum (TBR), must be a multiple of 4 bytes in length
APID Range:	0x3A0 - 0x3FF from LAT

Notes: None

2.2 Unit Identification

The LAT will use the following unit identification codes for external communications (via the 1553 bus) and internal communications (via the LATp bus):

<table>
<tr><th colspan="3">LAT Unit Identification Codes</th></tr>
<tr><th>LAT Unit</th><th>Description</th><th>LATp ID</th></tr>
<tr><td>Not Applicable</td><td>SIU(ext): SIU test port.</td><td>0x11 (TBR)</td></tr>
<tr><td rowspan="2">0x0</td><td>SIU(0): The SIU designated as unit 0. The LAT unit identifier of 0x0 will be used to address the active SIU on the 1553 bus. SIU 0 is active if it is the SIU that is responsive on the 1553 bus. The LATp address for SIU(0) is unique from SIU(1) and is designated as ID 18.</td><td>0x12</td></tr>
<tr><td>SIU(1): The SIU designated as unit 1. The LAT unit identifier of 0x0 will be used to address the active SIU on the 1553 bus. SIU 1 is active if it is the SIU that is responsive on the 1553 bus. The LATp address for SIU(1) is unique from SIU(0) and is designated as ID 19.</td><td>0x13</td></tr>
<tr><td>0x1</td><td>EPU(0): EPU designated as unit 0.</td><td>0x14</td></tr>
<tr><td>0x2</td><td>EPU(1): EPU designated as unit 1.</td><td>0x15</td></tr>
<tr><td>0x3</td><td>EPU(2): EPU designated as unit 2.</td><td>0x16</td></tr>
<tr><td>0x6-0xF</td><td>Not Assigned</td><td>Not Applicable</td></tr>
</table>

3 CCSDS PROTOCOL

3.0 Telecommand Packet

The LAT receives CCSDS telecommand packets as input from the SC, across the 1553 bus. The LAT can also send a limited set of telecommands to the SC, across the 1553 bus.

Packet Format

Each Telecommand packet contains three items:

- GLAST CCSDS Telecommand Packet Header
- Packet Command Data
- Packet Checksum (required by Spectrum Astro)

Header Format

The GLAST CCSDS telecommand packets have the standard 6-byte primary header, followed by a 2-byte secondary header. The GLAST CCSDS telecommand header layout is shown below.

Layout:

<table>
<tr><th>offset</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>0x00</td><td colspan="3">Version = 0</td><td>T=1</td><td>SH=1</td><td colspan="11">APID</td></tr>
<tr><td>0x02</td><td colspan="2">3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>0x04</td><td colspan="16">Packet Length</td></tr>
<tr><td>0x06</td><td>0</td><td colspan="15">Function Code</td></tr>
</table>

Fields:

APID - The CCSDS-packet application identifier.

Function Code - An indicator of the specific command action to perform.

Packet Length - The CCSDS-packet length. Indicates the length of application data, plus the two bytes of secondary header, minus 1.

Sequence Count - The CCSDS-packet sequence count. This running counter increments for each packet generated for a given application type (indicated by the "APID" member).

SF - The CCSDS-packet sequence flags ('00' = continuation packet in the middle of a sequence; '01' = first packet in a sequence; '10' = standalone packet that is not part of a sequence).

SH - The CCSDS-packet secondary-header flag. Always 1, indicating that all GLAST telemetry packets have secondary headers.

T - The CCSDS packet-type identifier ('0' indicates a telemetry packet; '1' indicates a telecommand packet).

Version - The CCSDS-packet version identifier. Always '0', indicating a Version 1 packet.

Telecommand Application Identifier (APID) Summary

Telecommand Receive Application ID's

Telecommand packets with telecommand receive application ID values designate commands or other input for which the LAT is the destination. LAT telecommand receive APID's are in the range 0x640-0x69F. In addition, the LAT also receives broadcast telecommands as input from the SC.

The LAT utilizes the APID range from 0x710-0x76F for internal LAT FSW master-task to slave-task communications. This APID range is used to address message queues associated with slave tasks. The LAT utilizes the APID range from 0x780-0x7DF for internal LAT FSW slave-task to master-task communications. This APID range is used to address message queues associated with master tasks. These APIDs are for internal communication use only and are not recognized by any LAT external system. Telecommands from the ground (across the 1553 bus) cannot utilize these APID ranges.

Telecommand Receive APIDs			
APID	**Description**	**M -> S**	**S -> M**
0x640	Boot operational telecommands	N/A	N/A
0x641	File Load telecommands	0x711	0x781
0x642	Memory Load telecommands	0x712	0x782
0x643	File Dump telecommands	0x713	0x783
0x644	Memory Dump telecommands	0x714	0x784
0x645	Task Management telecommands	0x715	0x785
0x646	Diagnostic telecommands	0x716	0x786
0x647	Not Assigned	N/A	N/A
0x648	Front end primitive telecommands	N/A	N/A
0x649	Not Assigned	N/A	N/A
0x650	Science operations telecommands	N/A	N/A
0x650-0x65F	Not Assigned	N/A	N/A
0x660	GBM alert telecommands	N/A	N/A
0x661	SC repoint request reply	N/A	N/A
0x663-0x69F	Not Assigned	N/A	N/A
0x662	SC load shed notification	N/A	N/A
0x701	SC broadcast telecommands	N/A	N/A

Telecommand Transmit Application ID’s

Telecommand packets with telecommand transmit application ID values designate commands or other input for which the LAT is the source. The APID assignments for such telecommands depend on the destination. The SC destination is designated by APID’s in the range 0x600-0x63F.

Telecommand Transmit APIDs	
APID	**Description**
0x600-0x63F	SC repoint request telecommands

3.1 Telemetry Packet Summary

The LAT outputs CCSDS telemetry packets to the SC on both the 1553 bus and the high-speed science data interface.

Packet Format

Each Telemetry packet contains two items:

- GLAST CCSDS Telecommand Packet Header
- Packet Command Data

Header Format

The LAT CCSDS telemetry packets have the standard 6-byte primary header, followed by an 8-byte secondary header specific to the LAT. The LAT CCSDS telemetry header layout is shown below.

Layout:

offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0x00	Version = 0			T=0	SH=1	APID										
0x02	SF		Sequence Count													
0x04	Packet Length: TBD															
0x06	Timestamp Seconds MSW															
0x08	Timestamp Seconds LSW															
0x0A	Timestamp Sub-Seconds MSW															
0x0C	Timestamp Sub-Seconds LSW															

APID - The CCSDS-packet application identifier.

Packet Length - The CCSDS-packet length. Indicates the length of application data, plus the two bytes of secondary header, minus 1.

Sequence Count - The CCSDS-packet sequence count. This running counter increments for each packet generated for a given application type (indicated by the "APID" member).

SF - The CCSDS-packet sequence flags ('00' = continuation packet in the middle of a sequence; '01' = first packet in a sequence; '10' = standalone packet that is not part of a sequence).

SH - The CCSDS-packet secondary-header flag. Always 1, indicating that all GLAST telemetry packets have secondary headers.

T - The CCSDS packet-type identifier ('0' indicates a telemetry packet; '1' indicates a telecommand packet).

Timestamp - Time is represented as a 64-bit offset from a time epoch. The time epoch for LAT and the spacecraft bus is 00:00:00.0 hours of January 1st, 2001. That is the midnight between December 31st, 2000 and January 1st, 2001. Timestamp Seconds represents the number of elapsed seconds since the epoch. Timestamp Sub-Seconds represents the number of microseconds elapsed since the last second.

Version - The CCSDS-packet version identifier. Always '0', indicating a Version 1 packet.

Telemetry Application Identifier (APID) Summary

Housekeeping Telemetry Application ID's

The housekeeping telemetry packets provide critical status information about the current state of the LAT. The SC treats housekeeping telemetry packets specially, so that the packets are downlinked on the real-time channel.

Housekeeping Telemetry APIDs	
APID	**Description**
0x200	Boot housekeeping telemetry
0x20F	Communication test housekeeping telemetry

Diagnostic Telemetry Appication ID's

Diagnostic telemetry packets provide status responses to particular commands or internal states of the LAT. The diagnostic telemetry may report general success or error reports, in response to telecommands. The diagnostic telemetry may also report data back in response to a telecommand that requests a dump or explicit status of some kind. Internal errors encountered during normal processing will also generate diagnostic telemetry error packets.

Diagnostic Telemetry APIDs	
APID	**Description**
0x260	File data dump telemetry
0x261	File directory dump telemetry
0x262	File system dump telemetry
0x263	Memory data dump telemetry
0x264	Memory symbol lookup telemetry
0x265	Memory pool status dump telemetry
0x266	Task status dump telemetry
0x26F	Communication test diagnostic telemetry
0x270	TEM register read telemetry
0x271	GTIC register read telemetry
0x272	GCCC register read telemetry
0x273	GCRC register read telemetry
0x274	GCFE register read telemetry
0x275	GTCC register read telemetry
0x276	GTRC register read telemetry
0x277	GTFE register read telemetry
0x278	AEM register read telemetry
0x279	GARC register read telemetry
0x27A	GAFE register read telemetry

Alert Telemetry Appication ID's

Alert Telemetry APIDs	
APID	**Description**
0x34F	Communication test alert telemetry

Science Telemetry Appication ID's

Science Telemetry APIDs	
APID	**Description**
0x3AF	Communication test science telemetry

4 PACKAGES

5 FILE Package

5.0 Overview

The FILE package contains routines that are specific to the file system.

The package supports the following functions:

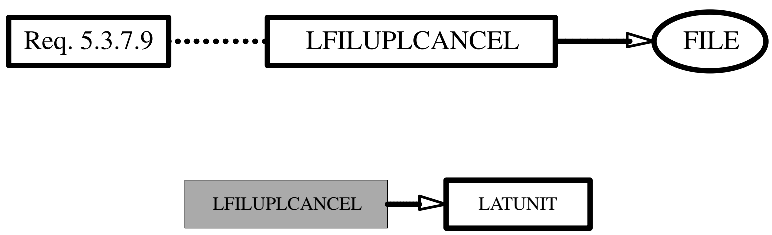
- RAD750 boot and crate initialization

5.1 Command Packets

ITOS Mnemonic: `LFILUPLCANCEL`

5.1.0 `LFILUPLCANCEL` (1601/0x641:1)

Context:



Description:

"File Upload Cancel" Telecommand Packet

This command cancels a file upload and resets the file upload state machine to the START state, regardless of the state when the command is received. A new LFILUPLSTART command is neede to begin a new file upload.

Layout:

File Upload Cancel (LFILUPLCANCEL) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x641:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=16**															
006	**0**	**Function Code=1**														
008	**LATUNIT:** latUnit															
00A	**Packet Checksum**															

```
LFILUPLCANCEL (pkt)
│  LAT Unit ID latUnit (fld)

typedef struct _FILE_LFILUPLCANCEL_Pkt {

    /* CCSDS header */
    char                 hdr[8];
```

```
    /* Command payload */
    FILE_LFILUPLCANCEL_Prm pay;

} FILE_LFILUPLCANCEL_Pkt;
```

Fields:

`LATUNIT` **(5.2.5)**

Attribute(s):

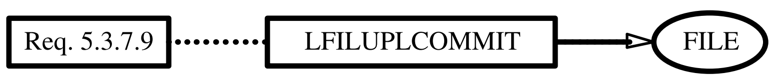
Brief: "LAT Unit ID"

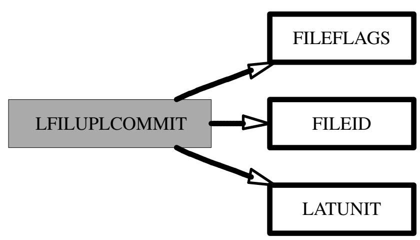
Instance(s): `latUnit`

ITOS Mnemonic: `LFILUPLCOMMIT`

5.1.1 `LFILUPLCOMMIT` (1601/0x641:2)

Context:





Description:

"File Upload Commit" Telecommand Packet

This command indicates that the loaded LFILUPLDATA file data contents should be written to storage. The FILEID paramter indicates the storage location. The complete file data set is first validated before successfully entering the COMMIT state. The validate only flag in the FILEFLAGS parameter indicates that validations errors shoud be reported, but that the file upload state machine will remain in the LOAD state regardless of the validation outcome.

Layout:

File Upload Commit (`LFILUPLCOMMIT`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x641:2**										
002	**SF=3**		**Sequence Count**													

File Upload Commit (LFILUPLCOMMIT) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
004	**Packet Length=64**															
006	**0**	**Function Code=2**														
008	**LATUNIT:** `latUnit`															
00A	**FILEFLAGS:** `fileFlags`															
00C	**FILEID:** `fileId`															
010	**Packet Checksum**															

```
LFILUPLCOMMIT (pkt)
|  LAT Unit ID latUnit (fld)
|  File Commit Flags fileFlags (fld)
|  File Storage ID fileId (fld)

typedef struct _FILE_LFILUPLCOMMIT_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    FILE_LFILUPLCOMMIT_Prm pay;

} FILE_LFILUPLCOMMIT_Pkt;
```

Fields:

`FILEFLAGS` **(5.2.1)**

Attribute(s):

Brief: "File Commit Flags"

Instance(s): `fileFlags`

`FILEID` **(5.2.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(5.2.5)**

Attribute(s):

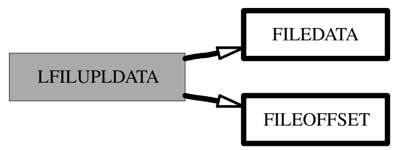
Brief: "LAT Unit ID"

Instance(s): `latUnit`

ITOS Mnemonic: `LFILUPLDATA`

5.1.2 `LFILUPLDATA` (1601/0x641:3)

Context:



Description:

"File Upload Data" Telecommand Packet

Each LFILUPLDATA packet contains a portion of the file data being loaded. The FILEOFFSET parameter is the offset in bytes from the beginning of the file which the first FILEDATA byte in this packet represents. The LFILEUPLDATA packets may be loaded in any order with respect to the offset and may be re-loaded ad many times as wished .

Layout:

File Upload Data (`LFILUPLDATA`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x641:3**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=416**															
006	**0**	**Function Code=3**														
008	**FILEOFFSET:** `fileOffset`															
00C	**FILEDATA:** `fileData[0-47]`															
03C	**Packet Checksum**															

```
LFILUPLDATA (pkt)
|  File Data Offset fileOffset (fld)
|  File Data fileData[0-47] (fld)

typedef struct _FILE_LFILUPLDATA_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    FILE_LFILUPLDATA_Prm pay;
```

```
} FILE_LFILUPLDATA_Pkt;
```

Fields:

`FILEDATA` **(5.2.0)**

Attribute(s):

Brief: "File Data"

Instance(s): `fileData`

`FILEOFFSET` **(5.2.3)**

Attribute(s):

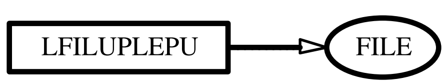
Brief: "File Data Offset"

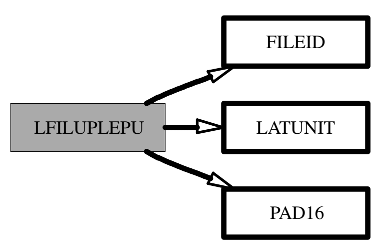
Instance(s): `fileOffset`

ITOS Mnemonic: `LFILUPLEPU`

5.1.3 `LFILUPLEPU` (1601/0x641:4)

Context:





Description:

"File Upload to EPU" Telecommand Packet

Layout:

File Upload to EPU (`LFILUPLEPU`) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x641:4**										

<table>
<tr><th colspan="17">File Upload to EPU (LFILUPLEPU)
Telecommand Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>002</td><td colspan="2">SF=3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=64</td></tr>
<tr><td>006</td><td>0</td><td colspan="15">Function Code=4</td></tr>
<tr><td>008</td><td colspan="16">LATUNIT: latUnit</td></tr>
<tr><td>00C</td><td colspan="16">FILEID: fileId</td></tr>
<tr><td>010</td><td colspan="16">Packet Checksum</td></tr>
</table>

```
LFILUPLEPU (pkt)
|  LAT Unit ID latUnit (fld)
|  File Storage ID fileId (fld)

typedef struct _FILE_LFILUPLEPU_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    FILE_LFILUPLEPU_Prm  pay;

} FILE_LFILUPLEPU_Pkt;
```

Fields:

`FILEID` **(5.2.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(5.2.5)**

Attribute(s):

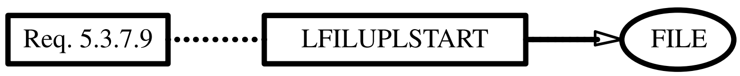
Brief: "LAT Unit ID"

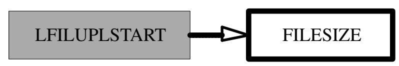
Instance(s): `latUnit`

ITOS Mnemonic: `LFILUPLSTART`

5.1.4 `LFILUPLSTART` (1601/0x641:0)

Context:





Description:

"File Upload Start" Telecommand Packet

Announces the start of a new file upload. The FILESIZE parameter sets the maximum offset for incoming LFILUPLDATA packets. Upon completion, the file upload state machine is in the LOAD state.

Layout:

File Upload Start (LFILUPLSTART)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x641:0										
002	SF=3		Sequence Count													
004	Packet Length=32															
006	0	Function Code=0														
008	FILESIZE: fileSize															
00C	Packet Checksum															

```
LFILUPLSTART (pkt)
|  File Size fileSize (fld)

typedef struct _FILE_LFILUPLSTART_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    FILE_LFILUPLSTART_Prm pay;

} FILE_LFILUPLSTART_Pkt;
```

Fields:

`FILESIZE` **(5.2.4)**

Attribute(s):

Brief: "File Size"

Instance(s): `fileSize`

5.2 Command Fields

5.2.0 `FILEDATA` (File Data) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`LFILUPLDATA (5.1.2)`

5.2.1 `FILEFLAGS` (File Commit Flags) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LFILUPLCOMMIT (5.1.1)`

5.2.2 `FILEID` (File Storage ID) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LFILUPLCOMMIT (5.1.1)`

5.2.3 `FILEOFFSET` (File Data Offset) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LFILUPLDATA (5.1.2)`

5.2.4 `FILESIZE` (File Size) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LFILUPLSTART (5.1.4)`

5.2.5 `LATUNIT` (LAT Unit ID) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LFILUPLCANCEL (5.1.0)`

6 ISIS Package

6.0 Overview

The ISIS package contains routines that are specific to the Instrument to Spacecraft Interface Simulator.

6.1 Enumerations

6.1.0 `CMD_CNT_SEL` (Selects count to return as diagnostic telemetry) Enumeration

ITOS Mnemonic: `CMD_CNT_SEL`

Description:

Definition:

0 No-op command (`NO_OP`)

1 Count of attitude messages from spacecraft (`ATTITUDE`)

2 Count of ancillary messages from spacecraft (`ANCILLARY`)

3 Count time-tone messages from spacecraft (`TIME_TONE`)

Used by:

???

6.1.1 `EPU_ID` (Enumeration of the EPUs) Enumeration

ITOS Mnemonic: `EPU_ID`

Description:

Definition:

1 EPU 1 (`EDP_1`)

1 EPU 0 (`EPU_0`)

2 EPU 2 (`EPU_2`)

3 External EPU (`EPU_X`)

Used by:

???

6.1.2 `FILEDEVICE` (Code for file device) Enumeration

ITOS Mnemonic: `FILEDEVICE`

Description:

Definition:

0 (`BOOT`)

1 RAM Device (RAM)

2 SIB EEPROM Patition 0 (EE0)

3 SIB EEPROM Partition 1 (EE1)

4 TMP file system (TMP)

Used by:

???

6.1.3 ON_OFF_SELECTOR (Enumeration of the options for the simple on-off selector) Enumeration

ITOS Mnemonic: ON_OFF_SELECTOR

Description:

Definition:

0 Off (OFF)

1 On (ON)

Used by:

???

6.1.4 PDU_ID (Identifies a PDU) Enumeration

ITOS Mnemonic: PDU_ID

Description:

Definition:

0 PDU 0 (PDU_0)

1 PDU 1 (PDU_1)

Used by:

???

6.1.5 P_S_SELECTOR (Enumeration for the simple primary-secondary selector) Enumeration

ITOS Mnemonic: P_S_SELECTOR

Description:

Definition:

0 Primary (PRIMARY)

1 Secondary (SECONDARY)

Used by:

???

6.1.6 `SCIPATTYPE` (Science data generation pattern types) Enumeration

ITOS Mnemonic: `SCIPATTYPE`

Description:

Definition:

0 Increment by 1 starting with parameter (`INCREMENT`)

1 Constant value using parameter (`CONSTANT`)

2 Random value using parameter as seed (`RANDOM`)

3 Walking one with parameter as width (`WALK1`)

4 Transition 0 to 1 with parameter size (`TRANS01`)

5 Transition 1 to 0 with parameter size (`TRANS10`)

Used by:

???

6.1.7 `SIU_ID` (Enumeration of the possible SIU IDs) Enumeration

ITOS Mnemonic: `SIU_ID`

Description:

Definition:

0 SIU 0 (aka SIU 7 aka primary SIU) (`SIU_0`)

1 SIU 1 (aka SIU 11 aka secondary SIU) (`SIU_1`)

2 External SIU (`SIU_X`)

Used by:

???

6.2 Ranges

6.2.0 `bits_12_range` (Range for 12-bit fields) Range

ITOS Mnemonic: `bits_12_range`

Description:

Definition:

Limits 0 - 4095

Used by:

???

6.2.1 `cmd_cnt_range` (Range for command count selection) Range

ITOS Mnemonic: `cmd_cnt_range`

Description:

Definition:

Limits 0 - 3

Used by:

???

6.2.2 `epu_range` (EPU number range) Range

ITOS Mnemonic: `epu_range`

Description:

Definition:

Limits 0 - 7

Used by:

???

6.2.3 `on_off_range` (On and off selector range) Range

ITOS Mnemonic: `on_off_range`

Description:

Definition:

Limits 0 - 1

Used by:

???

6.2.4 `p_s_range` (Primary and secondary selector range) Range

ITOS Mnemonic: `p_s_range`

Description:

Definition:

Limits 0 - 1

Used by:

???

6.2.5 `pdu_id_range` (PDU ID range) Range

ITOS Mnemonic: `pdu_id_range`

Description:

Definition:

Limits 0 - 1

Used by:

???

6.2.6 `siu_id_range` (SIU ID range) Range

ITOS Mnemonic: `siu_id_range`

Description:

Definition:

Limits 0 - 2

Used by:

???

6.2.7 `tem_mask_range` (TEM mask range) Range

ITOS Mnemonic: `tem_mask_range`

Description:

Definition:

Limits 0 - 65535

Used by:

???

7 ITC Package

7.0 Overview

The ITC package defines a common communications standard to unify communications between tasks on either the same or different CPUs. It also provides the services to build up tasks that are capable of communicating according to the standard.

The package supports the following functions:

- CPU internal communications/task frameworks

7.1 Telemetry Packets

7.1.0 `CmdConfirm` (720/0x2D0)

Context:

Description:

"Response to command from spacecraft" Telemetry Packet

Command confirmation telemetry packet. When executing spacecraft commands, ITC will autogenerate a diagnostic packet, detailing where and when a command is executed as well as the return code for the execution. To identify the command, ITC reflects the command in toto.

Layout:

Response to command from spacecraft (`CmdConfirm`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x2D0**										
002	**SF**		**Sequence Count**													
004	**Packet Length=41**															
006	**Timestamp**															
00E	**ITC_NodeID:** `nid`								**ITC_TaskID:** `tid`							
010	**Time_msui:** `deq_msui`															
014	**Time_lsui:** `deq_lsui`															
018	**Time_msui:** `exe_msui`															
01C	**Time_lsui:** `exe_lsui`															
020	**Status:** `status`															
024	**Drop:** `drp`															
026	**Pad2:** `pad`															
028	**CmdHeader:** `cmd_hdr`															

```
CmdConfirm (pkt)
|  ITC node ID nid (fld)
|  ITC task ID tid (fld)
|  Time (most significant 32 bits) deq_msui (fld)
|  Time (least significant 32 bits) deq_lsui (fld)
|  Time (most significant 32 bits) exe_msui (fld)
|  Time (least significant 32 bits) exe_lsui (fld)
```

```
│  MSG status code status (fld)
│  Number of dropped confirmations drp (fld)
│  Pad two bytes pad (fld)
│  CCSDS telecommand header cmd_hdr (str)
│  │  CCSDS telecommand header (bytes 0-1) hdr0 (bf)
│  │  │  CCSDS version ver (0-2)
│  │  │  CCSDS command bit isCmd (3)
│  │  │  CCSDS secondary header flag sec_hdr (4)
│  │  │  CCSDS application ID apid (5-15)
│  │  CCSDS telecommand header (bytes 2-3) hdr1 (bf)
│  │  │  CCSDS sequencing bits seq_flg (0-1)
│  │  │  CCSDS sequencing count seq_cnt (2-15)
│  │  CCSDS telecommand packet length cmd_len (fld)
│  │  CCSDS telecommand header (bytes 6-7) hdr3 (bf)
│  │  │  Padding fnc_pad (0)
│  │  │  CCSDS telecommand function code fnc (1-15)
```

```
typedef struct _ITC_CmdConfirm_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* ITC node ID */
    char                nid;

    /* ITC task ID */
    char                tid;

    /* Time (most significant 32 bits) */
    unsigned int        deq_msui;

    /* Time (least significant 32 bits) */
    unsigned int        deq_lsui;

    /* Time (most significant 32 bits) */
    unsigned int        exe_msui;

    /* Time (least significant 32 bits) */
    unsigned int        exe_lsui;

    /* MSG status code */
    unsigned int        status;

    /* Number of dropped confirmations */
    unsigned short      drp;

    /* Pad two bytes */
    unsigned short      pad;

    /* CCSDS telecommand header */
    ITC_CmdHeader       cmd_hdr;

} ITC_CmdConfirm_Tlm;
```

Fields:

`CmdHeader` **(7.2.0)**

Attribute(s):

Brief: "CCSDS telecommand header"

Instance(s): `cmd_hdr`

CCSDS telecommand header

`Drop` **(7.4.2)**

Attribute(s):

Brief: "Number of dropped confirmations"

Instance(s): `drp`

Should memory constraints result in dropped command confirmation packets, the count of dropped packets (on a per task basis) is accumulated and returned in the next successful command confirmation.

`ITC_NodeID` **(7.4.5)**

Attribute(s):
dsc: ITC_NODEID (7.5.0)

Brief: "ITC node ID"

Instance(s): `nid`

ITC identifies nodes on the communications network by number. These are the numbers.

`ITC_TaskID` **(7.4.6)**

Attribute(s):
dsc: ITC_TASKID (7.5.1)

Brief: "ITC task ID"

Instance(s): `tid`

ITC identifies tasks by number. These are the numbers.

`Pad2` **(7.4.7)**

Attribute(s):

Brief: "Pad two bytes"

Instance(s): `pad`

Two bytes of padding

`Status` (7.4.11)

Attribute(s):

Brief: "MSG status code"

Instance(s): `status`

The return code from the routine that executes the command.

`Time_lsui` (7.4.12)

Attribute(s):

Brief: "Time (least significant 32 bits)"

Instance(s): `deq_lsui`, `exe_lsui`

Lower half of a WCT_time object (an 8-byte integer measuring nanoseconds since the LAT epoch).

`Time_msui` (7.4.13)

Attribute(s):

Brief: "Time (most significant 32 bits)"

Instance(s): `deq_msui`, `exe_msui`

Upper half of a WCT_time object (an 8-byte integer measuring nanoseconds since the LAT epoch).

7.2 Telemetry Structs

7.2.0 `CmdHeader` (CCSDS telecommand header) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 64 bits (8 bytes)

Description:

CCSDS telecommand header

CCSDS telecommand header (`CmdHeader`) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	Hdr0: `hdr0`															
002	Hdr1: `hdr1`															
004	Cmd_len: `cmd_len`															
006	Hdr3: `hdr3`															

```
CmdHeader (str)
|  CCSDS telecommand header (bytes 0-1) hdr0 (bf)
|  |  CCSDS version ver (0-2)
|  |  CCSDS command bit isCmd (3)
|  |  CCSDS secondary header flag sec_hdr (4)
|  |  CCSDS application ID apid (5-15)
|  CCSDS telecommand header (bytes 2-3) hdr1 (bf)
|  |  CCSDS sequencing bits seq_flg (0-1)
|  |  CCSDS sequencing count seq_cnt (2-15)
|  CCSDS telecommand packet length cmd_len (fld)
|  CCSDS telecommand header (bytes 6-7) hdr3 (bf)
|  |  Padding fnc_pad (0)
|  |  CCSDS telecommand function code fnc (1-15)
```

```
typedef struct _ITC_CmdHeader {

    /* CCSDS telecommand header (bytes 0-1) */
    ITC_Hdr0             hdr0;

    /* CCSDS telecommand header (bytes 2-3) */
    ITC_Hdr1             hdr1;

    /* CCSDS telecommand packet length */
    unsigned short       cmd_len;

    /* CCSDS telecommand header (bytes 6-7) */
    ITC_Hdr3             hdr3;

} ITC_CmdHeader;
```

Fields:

`Cmd_len` **(7.4.1)**

Attribute(s):

Brief: "CCSDS telecommand packet length"

Instance(s): `cmd_len`

CCSDS packet length (CCSDS definition, which is a little wierd).

`Hdr0` **(7.3.0)**

Attribute(s):

Brief: "CCSDS telecommand header (bytes 0-1)"

Instance(s): `hdr0`

Bitfields for the bytes 0-1 of a standard CCSDS telecommand header

`Hdr1` **(7.3.1)**

Attribute(s):

Brief: "CCSDS telecommand header (bytes 2-3)"

Instance(s): `hdr1`

Bitfileds for bytes 2-3 of a standard CCSDS telecommand header

`Hdr3` **(7.3.2)**

Attribute(s):

Brief: "CCSDS telecommand header (bytes 6-7)"

Instance(s): `hdr3`

Bitfields for bytes 6-7 of a standard CCSDS telecommand header, where a two-byte secondary header containing a function code has been defined.

Used by:

`CmdConfirm (7.1.0)`

7.3 Telemetry Bitfields

7.3.0 `Hdr0` (CCSDS telecommand header (bytes 0-1)) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfields for the bytes 0-1 of a standard CCSDS telecommand header

Start	Stop	Size	Item Type and Name
0	2	3	`Version ver`
3	3	1	`isCmd isCmd`
4	4	1	`Sec_hdr sec_hdr`
5	15	11	`Apid apid`

```
Hdr0 (bf)
|  CCSDS version ver (0-2)
|  CCSDS command bit isCmd (3)
|  CCSDS secondary header flag sec_hdr (4)
|  CCSDS application ID apid (5-15)
```

```
typedef struct _ITC_Hdr0_Bfl {

    #if ENDIANNESS_IS_BIG

    /* CCSDS version */
    unsigned short       ver:3;

    /* CCSDS command bit */
    unsigned short       isCmd:1;

    /* CCSDS secondary header flag */
    unsigned short       sec_hdr:1;

    /* CCSDS application ID */
    unsigned short       apid:11;

    #elif ENDIANNESS_IS_LITTLE

    /* CCSDS application ID */
    unsigned short       apid:11;

    /* CCSDS secondary header flag */
    unsigned short       sec_hdr:1;

    /* CCSDS command bit */
    unsigned short       isCmd:1;

    /* CCSDS version */
    unsigned short       ver:3;
```

```
    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) ITC_Hdr0_Bfl;

typedef union _ITC_Hdr0 {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _ITC_Hdr0_Bfl           bf;

} ITC_Hdr0;
```

Fields:

`Apid` **(7.4.0)**

Attribute(s):

Brief: "CCSDS application ID"

Instance(s): `apid`

Standard CCSDS application identifier.

`isCmd` **(7.4.15)**

Attribute(s):

Brief: "CCSDS command bit"

Instance(s): `isCmd`

Set for telecommands, clear for telemetry.

`Sec_hdr` **(7.4.8)**

Attribute(s):

Brief: "CCSDS secondary header flag"

Instance(s): `sec_hdr`

Set when a CCSDS packet includes a secondary header. Clear otherwise. All LAT telemetry/telecommands carry a secondary header.

`Version` **(7.4.14)**

Attribute(s):

Brief: "CCSDS version"

Instance(s): `ver`

Standard CCSDS version code

Used by:

`CmdHeader (7.2.0)`

7.3.1 `Hdr1` (CCSDS telecommand header (bytes 2-3)) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Bitfileds for bytes 2-3 of a standard CCSDS telecommand header

Start	Stop	Size	Item Type and Name
0	1	2	`Seq_flg seq_flg`
2	15	14	`Seq_cnt seq_cnt`

```
Hdr1 (bf)
|  CCSDS sequencing bits seq_flg (0-1)
|  CCSDS sequencing count seq_cnt (2-15)
```

```
typedef struct _ITC_Hdr1_Bfl {

    #if ENDIANNESS_IS_BIG

    /* CCSDS sequencing bits */
    unsigned short       seq_flg:2;

    /* CCSDS sequencing count */
    unsigned short       seq_cnt:14;

    #elif ENDIANNESS_IS_LITTLE

    /* CCSDS sequencing count */
    unsigned short       seq_cnt:14;

    /* CCSDS sequencing bits */
    unsigned short       seq_flg:2;

    #else
```

```
    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) ITC_Hdr1_Bfl;

typedef union _ITC_Hdr1 {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _ITC_Hdr1_Bfl            bf;

} ITC_Hdr1;
```

Fields:

`Seq_cnt` **(7.4.9)**

Attribute(s):

Brief: "CCSDS sequencing count"

Instance(s): `seq_cnt`

Standard CCSDS sequencing count (incrementing per APID).

`Seq_flg` **(7.4.10)**

Attribute(s):

Brief: "CCSDS sequencing bits"

Instance(s): `seq_flg`

Standard CCSDS sequencing bits.

Used by:

`CmdHeader (7.2.0)`

7.3.2 `Hdr3` (CCSDS telecommand header (bytes 6-7)) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfields for bytes 6-7 of a standard CCSDS telecommand header, where a two-byte secondary header containing a function code has been defined.

Start	Stop	Size	Item Type and Name
0	0	1	`Fnc_pad fnc_pad`
1	15	15	`Fnc_code fnc`

```
Hdr3 (bf)
|  Padding fnc_pad (0)
|  CCSDS telecommand function code fnc (1-15)
```

```
typedef struct _ITC_Hdr3_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Padding */
    unsigned short       fnc_pad:1;

    /* CCSDS telecommand function code */
    unsigned short       fnc:15;

    #elif ENDIANNESS_IS_LITTLE

    /* CCSDS telecommand function code */
    unsigned short       fnc:15;

    /* Padding */
    unsigned short       fnc_pad:1;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) ITC_Hdr3_Bfl;

typedef union _ITC_Hdr3 {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _ITC_Hdr3_Bfl           bf;

} ITC_Hdr3;
```

Fields:

`Fnc_code` **(7.4.3)**

Attribute(s):

Brief: "CCSDS telecommand function code"

Instance(s): `fnc`

CCSDS telecommand function code

`Fnc_pad` **(7.4.4)**

Attribute(s):

Brief: "Padding"

Instance(s): `fnc_pad`

Function codes are not allowed to touch the most significnat bit of the 16-bit function code. This padding places the bit off limits.

Used by:

`CmdHeader (7.2.0)`

7.4 Telemetry Fields

7.4.0 `Apid` (CCSDS application ID) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	11 bits

Description:

Standard CCSDS application identifier.

Used by:

`Hdr0 (7.3.0)`

7.4.1 `Cmd_len` (CCSDS telecommand packet length) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

CCSDS packet length (CCSDS definition, which is a little wierd).

Used by:

`CmdHeader (7.2.0)`

7.4.2 `Drop` (Number of dropped confirmations) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Should memory constraints result in dropped command confirmation packets, the count of dropped packets (on a per task basis) is accumulated and returned in the next successful command confirmation.

Used by:

`CmdConfirm (7.1.0)`

7.4.3 `Fnc_code` (CCSDS telecommand function code) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	15 bits

Description:

CCSDS telecommand function code

Used by:

`Hdr3 (7.3.2)`

7.4.4 `Fnc_pad` (Padding) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Function codes are not allowed to touch the most significnat bit of the 16-bit function code. This padding places the bit off limits.

Used by:

`Hdr3 (7.3.2)`

7.4.5 `ITC_NodeID` (ITC node ID) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

ITC identifies nodes on the communications network by number. These are the numbers.

Used by:

`CmdConfirm (7.1.0)`

7.4.6 `ITC_TaskID` (ITC task ID) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

ITC identifies tasks by number. These are the numbers.

Used by:

`CmdConfirm (7.1.0)`

7.4.7 `Pad2` (Pad two bytes) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Two bytes of padding

Used by:

`CmdConfirm (7.1.0)`

7.4.8 `Sec_hdr` (CCSDS secondary header flag) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Set when a CCSDS packet includes a secondary header. Clear otherwise. All LAT telemetry/telecommands carry a secondary header.

Used by:

`Hdr0 (7.3.0)`

7.4.9 `Seq_cnt` (CCSDS sequencing count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	14 bits

Description:

Standard CCSDS sequencing count (incrementing per APID).

Used by:

`Hdr1 (7.3.1)`

7.4.10 `Seq_flg` (CCSDS sequencing bits) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	2 bits

Description:

Standard CCSDS sequencing bits.

Used by:

`Hdr1 (7.3.1)`

7.4.11 `Status` (MSG status code) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The return code from the routine that executes the command.

Used by:

`CmdConfirm (7.1.0)`

7.4.12 `Time_lsui` (Time (least significant 32 bits)) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Lower half of a WCT_time object (an 8-byte integer measuring nanoseconds since the LAT epoch).

Used by:

`CmdConfirm (7.1.0)`

7.4.13 `Time_msui` (Time (most significant 32 bits)) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Upper half of a WCT_time object (an 8-byte integer measuring nanoseconds since the LAT epoch).

Used by:

`CmdConfirm (7.1.0)`

7.4.14 `Version` (CCSDS version) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	3 bits

Description:

Standard CCSDS version code

Used by:

`Hdr0 (7.3.0)`

7.4.15 `isCmd` (CCSDS command bit) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Set for telecommands, clear for telemetry.

Used by:

`Hdr0 (7.3.0)`

7.5 Discretes

7.5.0 ITC_NODEID (Discrete list of ITC nodes) Discrete

ITOS Mnemonic: ITC_NODEID

Description:

Discrete list of ITC nodes

Definition:

-1 Unknown node (guard value) (ITC_NID_UNKNOWN)

0 SIU (ITC_NID_SIU)

1 EPU 0 (ITC_NID_EPU0)

2 EPU 1 (ITC_NID_EPU1)

3 EPU 2 (ITC_NID_EPU2)

4 EPU 3 (external crate) (ITC_NID_EPU3)

5 Science Data Interface (to Solid State Recorder) (ITC_NID_SDI)

6 Spacecraft (ITC_NID_SC)

7 CPU broadcast class (ITC_NID_BCST)

Used by:

CmdConfirm (7.1.0)

7.5.1 ITC_TASKID (Discrete list of ITC task IDs) Discrete

ITOS Mnemonic: ITC_TASKID

Description:

Discrete list of ITC task IDs

Definition:

-1 Unknown task (guard value) (ITC_TID_UNKNOWN)

0 The anonymous task (ITC_TID_ANON)

1 LAT computer manager task (ITC_TID_LCM)

2 LAT file system master task (ITC_TID_LFS_M)

3 LAT file system slave task (LAT_TID_LFS_S)

4 LAT housekeeping master task (LAT_TID_LHK_M)

5 LAT housekeeping slave task (ITC_TID_LHK_S)

6 LAT instrument manager master task (`LAT_TID_LIM_M`)

7 LAT instrument manager slave task (`LAT_TID_LIM_S`)

8 LAT spacecraft messages master task (`LAT_TID_LSM_M`)

9 LAT spacecraft messages slave task (`ITC_TID_LSM_S`)

10 LAT software watchdog master task (`ITC_TID_LSW_M`)

11 LAT software watchdog slave task (`ITC_TID_LSW_S`)

12 LAT charge injection calibration master task (`ITC_TID_LCI_M`)

13 LAT charge injection calibration slave task (`ITC_TID_LCI_S`)

20 ISIS command task (`ITC_TID_ICT`)

30 LCB Tx service task (`ITS_SID_LCS`)

31 CTDB Tx service task (`ITC_SID_CTS`)

32 LCB Rx driver callback (`ITC_LID_LCX`)

33 CTDB Rx driver callback (`ITC_LID_CTX`)

34 ITC hook into message task (`ITC_LID_MSG`)

Used by:

`CmdConfirm (7.1.0)`

8 LCM Package

8.0 Overview

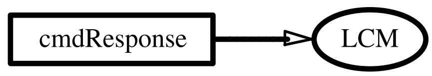
The LCM package handles the management of a single LAT computer.

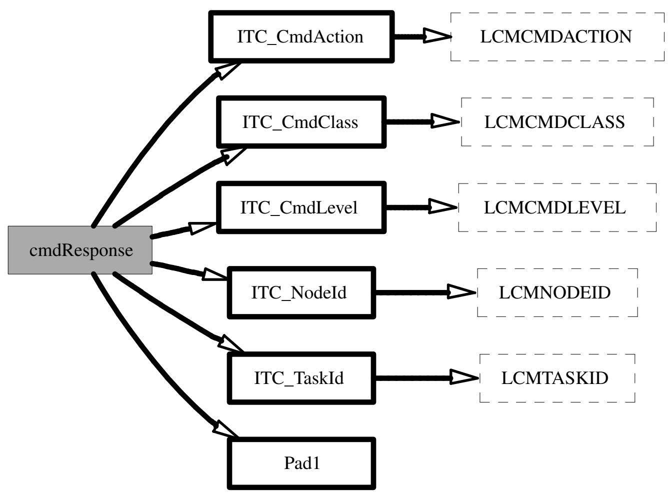
8.1 Command Packets

ITOS Mnemonic: `CmdResponse`

8.1.0 `cmdResponse` (1685/0x695:1)

Context:





Description:

"Change task command confirmation level" Telecommand Packet

Change the spacecraft command confirmation level of a task

Layout:

Change task command confirmation level (`cmdResponse`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x695:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=9**															
006	**0**	**Function Code=1**														
008	**ITC_NodeId:** `nid`								**ITC_TaskId:** `tid`							

Change task command confirmation level (cmdResponse) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00A	**ITC_CmdClass:** scb								**ITC_CmdAction:** fwd							
00C	**ITC_CmdLevel:** lvl								**Pad1:** pad							
00E	**Packet Checksum**															

```
cmdResponse (pkt)
|  ITC node ID nid (fld)
|  ITC task ID tid (fld)
|  Spacecraft command class (normal or broadcast) scb (fld)
|  Actions task can take (forward or execute) fwd (fld)
|  Response level lvl (fld)
|  One byte padding pad (fld)
```

```
typedef struct _LCM_cmdResponse_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LCM_cmdResponse_Prm  pay;

} LCM_cmdResponse_Pkt;
```

Fields:

`ITC_CmdAction` **(8.2.0)**

Attribute(s):

Brief: "Actions task can take (forward or execute)"

Instance(s): `fwd`

`ITC_CmdClass` **(8.2.1)**

Attribute(s):

Brief: "Spacecraft command class (normal or broadcast)"

Instance(s): `scb`

`ITC_CmdLevel` **(8.2.2)**

Attribute(s):

Brief: "Response level"

Instance(s): `lvl`

`ITC_NodeId` **(8.2.3)**

Attribute(s):

Brief: "ITC node ID"

Instance(s): `nid`

`ITC_TaskId` **(8.2.4)**

Attribute(s):

Brief: "ITC task ID"

Instance(s): `tid`

`Pad1` **(8.2.6)**

Attribute(s):

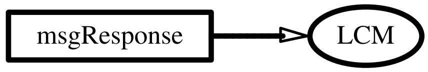
Brief: "One byte padding"

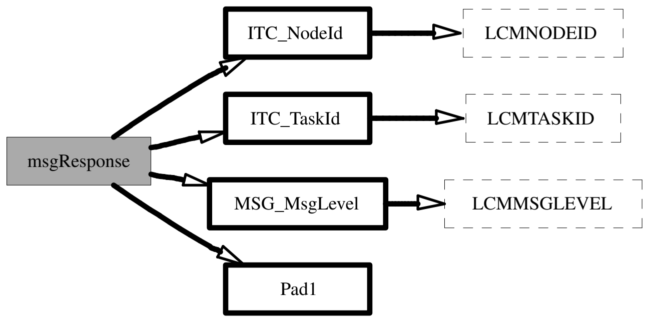
Instance(s): `pad`

ITOS Mnemonic: `MsgResponse`

8.1.1 `msgResponse` (1685/0x695:0)

Context:





Description:

"Change task messaging level" Telecommand Packet

Change the messaging level of a task

Layout:

Change task messaging level (`msgResponse`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x695:0**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=7**															
006	**0**	**Function Code=0**														
008	**ITC_NodeId:** `nid`								**ITC_TaskId:** `tid`							
00A	**MSG_MsgLevel:** `lvl`								**Pad1:** `pad`							
00C	**Packet Checksum**															

```
msgResponse (pkt)
|  ITC node ID nid (fld)
|  ITC task ID tid (fld)
|  Messaging level lvl (fld)
|  One byte padding pad (fld)

typedef struct _LCM_msgResponse_Pkt {

    /* CCSDS header */
    char                 hdr[8];
```

```
    /* Command payload */
    LCM_msgResponse_Prm  pay;

} LCM_msgResponse_Pkt;
```

Fields:

`ITC_NodeId` **(8.2.3)**

Attribute(s):

Brief: "ITC node ID"

Instance(s): `nid`

`ITC_TaskId` **(8.2.4)**

Attribute(s):

Brief: "ITC task ID"

Instance(s): `tid`

`MSG_MsgLevel` **(8.2.5)**

Attribute(s):

Brief: "Messaging level"

Instance(s): `lvl`

`Pad1` **(8.2.6)**

Attribute(s):

Brief: "One byte padding"

Instance(s): `pad`

8.2 Command Fields

8.2.0 `ITC_CmdAction` (Actions task can take (forward or execute)) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

`cmdResponse (8.1.0)`

8.2.1 `ITC_CmdClass` (Spacecraft command class (normal or broadcast)) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

`cmdResponse (8.1.0)`

8.2.2 `ITC_CmdLevel` (Response level) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

`cmdResponse (8.1.0)`

8.2.3 `ITC_NodeId` (ITC node ID) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

cmdResponse (8.1.0)

8.2.4 `ITC_TaskId` (ITC task ID) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

cmdResponse (8.1.0)

8.2.5 `MSG_MsgLevel` (Messaging level) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

msgResponse (8.1.1)

8.2.6 `Pad1` (One byte padding) Telecommand Field

Definition:

Alignment:	1 byte
C type:	char
Length:	8 bits (1 byte)

Description:

Used by:

cmdResponse (8.1.0)

9 LFS Package

9.0 Overview

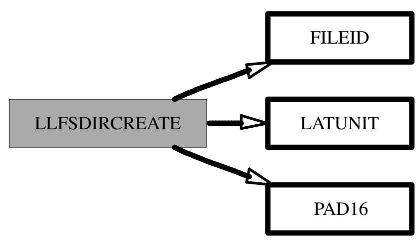
The LFS package provides the "standard" file operations (copy, delete, move, list directory, ...), plus a few "unusual" operations (file upload and file dump via spacecraft).

9.1 Command Packets

ITOS Mnemonic: `LLFSDIRCREATE`

9.1.0 `LLFSDIRCREATE` (1608/0x648:2)

Context:



Description:

"Directory Create" Telecommand Packet

Creates a directory specified by the FILEID path. Only the device and directory number portions of the file ID are used.

Layout:

Directory Create (LLFSDIRCREATE) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x648:2**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=64**															
006	**0**	**Function Code=2**														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `fileId`															
010	**Packet Checksum**															

```
LLFSDIRCREATE (pkt)
│  LAT Unit ID and Transaction ID latUnit (fld)
```

```
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSDIRCREATE_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LFS_LLFSDIRCREATE_Prm pay;

} LFS_LLFSDIRCREATE_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

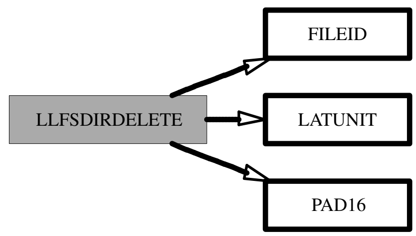
Brief: "16-bit padding"

Instance(s): `pad16`

ITOS Mnemonic: `LLFSDIRDELETE`

9.1.1 `LLFSDIRDELETE` (1608/0x648:3)

Context:



Description:

"Directory Delete" Telecommand Packet

Deletes a directory specified by the FILEID path. Only the device and directory number portions of the file ID are used.

Layout:

Directory Delete (`LLFSDIRDELETE`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x648:3**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=64**															
006	**0**	**Function Code=3**														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `fileId`															
010	**Packet Checksum**															

```
LLFSDIRDELETE (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSDIRDELETE_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
```

```
    LFS_LLFSDIRDELETE_Prm pay;
} LFS_LLFSDIRDELETE_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

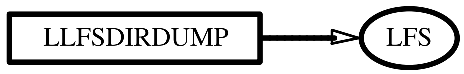
Brief: "16-bit padding"

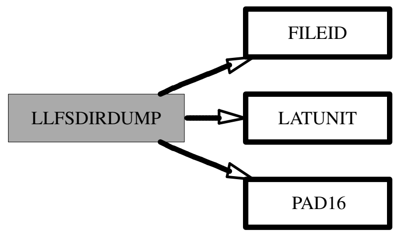
Instance(s): `pad16`

ITOS Mnemonic: `LLFSDIRDUMP`

9.1.2 `LLFSDIRDUMP` (1608/0x648:5)

Context:





Description:

"Directory Dump" Telecommand Packet

Dumps the contents of a directory specified by the FILEID path. Only the device and directory number portions of the file ID are used. If the directory value is '127', a series of LLFSROOTLIST telemetry packets are sent, one for each directory in the root directory of the device. Otherwise, a series of LLFSDIRLIST telemetry packets are sent, one for each file in the directory.

Layout:

Directory Dump (`LLFSDIRDUMP`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x648:5**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=64**															
006	**0**	**Function Code=5**														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `fileId`															
010	**Packet Checksum**															

```
LLFSDIRDUMP (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSDIRDUMP_Pkt {

    /* CCSDS header */
    char                 hdr[8];
```

```
    /* Command payload */
    LFS_LLFSDIRDUMP_Prm  pay;

} LFS_LLFSDIRDUMP_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

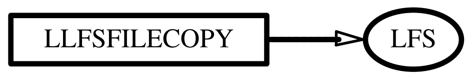
Brief: "16-bit padding"

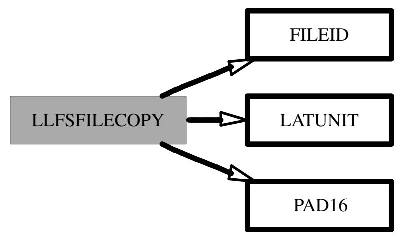
Instance(s): `pad16`

ITOS Mnemonic: `LLFSFILECOPY`

9.1.3 `LLFSFILECOPY` (1608/0x648:1)

Context:





Description:

"File Copy Local" Telecommand Packet

Copies a file locally on the same CPU. The FILEID values specify the source and destination file locations.

Layout:

File Copy Local (`LLFSFILECOPY`) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x648:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=96**															
006	**0**	**Function Code=1**														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `srcFileId`															
010	**FILEID:** `destFileId`															
014	**Packet Checksum**															

```
LLFSFILECOPY (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID srcFileId (fld) ... destFileId (fld)

typedef struct _LFS_LLFSFILECOPY_Pkt {

    /* CCSDS header */
    char                 hdr[8];
```

```
    /* Command payload */
    LFS_LLFSFILECOPY_Prm pay;

} LFS_LLFSFILECOPY_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `destFileId`, `srcFileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

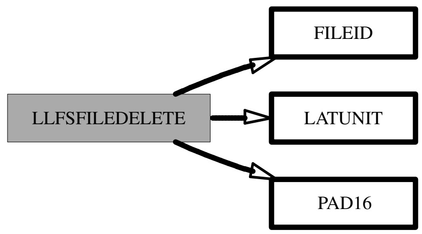
Brief: "16-bit padding"

Instance(s): `pad16`

ITOS Mnemonic: `LLFSFILEDELETE`

9.1.4 **LLFSFILEDELETE** (1608/0x648:0)

Context:



Description:

"File Delete" Telecommand Packet

Deletes the file specified by FILEID.

Layout:

File Delete (LLFSFILEDELETE) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x648:0**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=64**															
006	**0**	**Function Code=0**														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `fileId`															
010	**Packet Checksum**															

```
LLFSFILEDELETE (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSFILEDELETE_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LFS_LLFSFILEDELETE_Prm pay;
```

```
} LFS_LLFSFILEDELETE_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

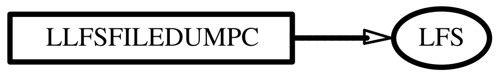
Brief: "16-bit padding"

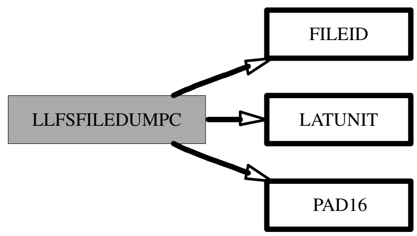
Instance(s): `pad16`

ITOS Mnemonic: `LLFSFILEDUMPC`

9.1.5 **LLFSFILEDUMPC** (1608/0x648:4)

Context:





Description:

"File Dump CTDB" Telecommand Packet

Dumps the contents of a file indicated by FILEID to the CTDB interface. The file data is sent as a series of LLFSDUMPTCTBD telemetery packets.

Layout:

File Dump CTDB (LLFSFILEDUMPC)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x648:4										
002	SF=3		Sequence Count													
004	Packet Length=64															
006	0	Function Code=4														
008	LATUNIT: latUnit															
00A	PAD16: pad16															
00C	FILEID: fileId															
010	Packet Checksum															

```
LLFSFILEDUMPC (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSFILEDUMPC_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
```

```
    LFS_LLFSFILEDUMPC_Prm pay;
} LFS_LLFSFILEDUMPC_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

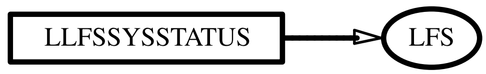
Brief: "16-bit padding"

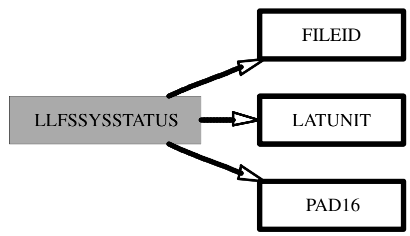
Instance(s): `pad16`

ITOS Mnemonic: `LLFSSYSSTATUS`

9.1.6 `LLFSSYSSTATUS` (1608/0x648:6)

Context:





Description:

"File System Status" Telecommand Packet

Report the current status of a file system partition indicated by FILEID. Only the device number portion of the file storage ID is used. The status is returned as a single LLFSSYSLIST telemetry packet.

Layout:

File System Status (`LLFSSYSSTATUS`)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x648:6										
002	SF=3		Sequence Count													
004	Packet Length=64															
006	0	Function Code=6														
008	**LATUNIT:** `latUnit`															
00A	**PAD16:** `pad16`															
00C	**FILEID:** `fileId`															
010	Packet Checksum															

```
LLFSSYSSTATUS (pkt)
|  LAT Unit ID and Transaction ID latUnit (fld)
|  16-bit padding pad16 (fld)
|  File Storage ID fileId (fld)

typedef struct _LFS_LLFSSYSSTATUS_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
```

```
    LFS_LLFSSYSSTATUS_Prm pay;

} LFS_LLFSSYSSTATUS_Pkt;
```

Fields:

`FILEID` **(9.2.0)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`LATUNIT` **(9.2.1)**

Attribute(s):

Brief: "LAT Unit ID and Transaction ID"

Instance(s): `latUnit`

`PAD16` **(9.2.2)**

Attribute(s):

Brief: "16-bit padding"

Instance(s): `pad16`

9.2 Command Fields

9.2.0 `FILEID` (File Storage ID) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDIRCREATE (9.1.0)`

9.2.1 `LATUNIT` (LAT Unit ID and Transaction ID) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LLFSDIRCREATE (9.1.0)`

9.2.2 `PAD16` (16-bit padding) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LLFSDIRCREATE (9.1.0)`

9.3 Telemetry Packets

9.3.0 **LLFSDIRLIST** (792/0x318)

Context:

Description:

"Directory Listing Report" Telemetry Packet

Lists one file entry in a LAT file system sub-directory. One of these packets is sent in response to a LLFSDIRDUMP telecommand for each file in the target directory.

Layout:

Directory Listing Report (LLFSDIRLIST) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x318										
002	SF		Sequence Count													
004	Packet Length=432															
006	Timestamp															
00E	**DUMPSTFLAGS:** `dmpFlags`															
010	**FILESTFLAGS:** `fileFlags`															
014	**FILESTID:** `fileId`															
018	**FILESTTIME:** `fileTime`															
01C	**FILESTSIZE:** `fileSize`															
020	**FILESTBLOCK:** `fileBlocks`															
024	**FILEHDR:** `fileHdr[0-31]`															

```
LLFSDIRLIST (pkt)
|  File Dump Flags dmpFlags (bf)
|  |  LAT Storage Unit latUnit (0-3)
|  |  Transaction ID xactId (4-15)
|  File Storage Flags fileFlags (bf)
|  |  File Storage Archive Flag archiveFlag (0)
|  |  File Storage Directory Flag dirFlag (1)
|  |  File Storage Read-Only Flag roFlag (2)
|  File Storage ID fileId (bf)
|  |  File ID Device Number dev (0-2)
|  |  File ID Directory Number dir (3-9)
|  |  File ID File Number file (10-31)
|  File Storage Update Time fileTime (fld)
|  File Storage Size fileSize (fld)
|  File Storage Blocks fileBlocks (fld)
|  File Header Data fileHdr[0-31] (fld)
```

```
typedef struct _LFS_LLFSDIRLIST_Tlm {

    /* Telemetry header */
    char                 hdr[14];
```

```
    /* File Dump Flags */
    LFS_DUMPSTFLAGS      dmpFlags;

    /* File Storage Flags */
    LFS_FILESTFLAGS      fileFlags;

    /* Byte pad */
    char                 byte_pad0[2];

    /* File Storage ID */
    LFS_FILESTID         fileId;

    /* File Storage Update Time */
    unsigned int         fileTime;

    /* File Storage Size */
    unsigned int         fileSize;

    /* File Storage Blocks */
    unsigned int         fileBlocks;

    /* File Header Data */
    unsigned char        fileHdr[32];

} LFS_LLFSDIRLIST_Tlm;
```

Fields:

`DUMPSTFLAGS` **(9.4.0)**

Attribute(s):

Brief: "File Dump Flags"

Instance(s): `dmpFlags`

`FILEHDR` **(9.5.4)**

Attribute(s):

Brief: "File Header Data"

Instance(s): `fileHdr`

`FILESTBLOCK` **(9.5.7)**

Attribute(s):

Brief: "File Storage Blocks"

Instance(s): `fileBlocks`

`FILESTFLAGS` **(9.4.1)**

Attribute(s):

Brief: "File Storage Flags"

Instance(s): `fileFlags`

`FILESTID` **(9.4.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`FILESTSIZE` **(9.5.11)**

Attribute(s):

Brief: "File Storage Size"

Instance(s): `fileSize`

`FILESTTIME` **(9.5.12)**

Attribute(s):

Brief: "File Storage Update Time"

Instance(s): `fileTime`

9.3.1 `LLFSDUMPCTDB` (795/0x31B)

Context:

Description:

"File Dump Data CTDB" Telemetry Packet

Contains a portion of a file data dump to the CTDB interface. The FILESTOFFSET and FILEDUMPSIZE members provide the offset into the file and size of the current packet's data contents (FILEDATA). The FILESTID gives the storage location of the file.

Layout:

File Dump Data CTDB (LLFSDUMPCTDB) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x31B**										
002	**SF**		**Sequence Count**													
004	**Packet Length=3232**															
006	**Timestamp**															
00E	**DUMPSTFLAGS:** `dmpFlags`															
010	**FILESTID:** `fileId`															
014	**FILESTOFFSET:** `fileOffset`															
018	**FILEDUMPSIZE:** `dmpSize`															
01C	**FILEDATA:** `fileData[0-389]`															

```
LLFSDUMPCTDB (pkt)
|  File Dump Flags dmpFlags (bf)
|  |  LAT Storage Unit latUnit (0-3)
|  |  Transaction ID xactId (4-15)
|  File Storage ID fileId (bf)
|  |  File ID Device Number dev (0-2)
|  |  File ID Directory Number dir (3-9)
|  |  File ID File Number file (10-31)
|  File Data Offset fileOffset (fld)
|  File Dump Current Size dmpSize (fld)
|  File Data Contents fileData[0-389] (fld)
```

```
typedef struct _LFS_LLFSDUMPCTDB_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* File Dump Flags */
    LFS_DUMPSTFLAGS      dmpFlags;

    /* File Storage ID */
    LFS_FILESTID         fileId;

    /* File Data Offset */
    unsigned int         fileOffset;

    /* File Dump Current Size */
    unsigned int         dmpSize;

    /* File Data Contents */
    unsigned char        fileData[390];

} LFS_LLFSDUMPCTDB_Tlm;
```

Fields:

`DUMPSTFLAGS` **(9.4.0)**

Attribute(s):

Brief: "File Dump Flags"

Instance(s): `dmpFlags`

`FILEDATA` **(9.5.0)**

Attribute(s):

Brief: "File Data Contents"

Instance(s): `fileData`

`FILEDUMPSIZE` **(9.5.3)**

Attribute(s):

Brief: "File Dump Current Size"

Instance(s): `dmpSize`

`FILESTID` **(9.4.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`FILESTOFFSET` **(9.5.9)**

Attribute(s):

Brief: "File Data Offset"

Instance(s): `fileOffset`

9.3.2 `LLFSROOTLIST` (793/0x319)

Context:

Description:

"Root Listing Report" Telemetry Packet

Lists one file entry in a LAT file system root directory. One of these packets is sent in response to a LLFSDIRDUMP (directory = '127') telecommand for each sub-directory in the target device.

Layout:

Root Listing Report (LLFSROOTLIST) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x319**										
002	**SF**		**Sequence Count**													
004	**Packet Length=176**															
006	**Timestamp**															
00E	**DUMPSTFLAGS:** `dmpFlags`															
010	**FILESTFLAGS:** `fileFlags`															
014	**FILESTID:** `fileId`															
018	**FILESTTIME:** `fileTime`															
01C	**FILESTSIZE:** `fileSize`															
020	**FILESTBLOCK:** `fileBlocks`															

```
LLFSROOTLIST (pkt)
|  File Dump Flags dmpFlags (bf)
|  |  LAT Storage Unit latUnit (0-3)
|  |  Transaction ID xactId (4-15)
|  File Storage Flags fileFlags (bf)
|  |  File Storage Archive Flag archiveFlag (0)
|  |  File Storage Directory Flag dirFlag (1)
|  |  File Storage Read-Only Flag roFlag (2)
|  File Storage ID fileId (bf)
|  |  File ID Device Number dev (0-2)
|  |  File ID Directory Number dir (3-9)
|  |  File ID File Number file (10-31)
|  File Storage Update Time fileTime (fld)
|  File Storage Size fileSize (fld)
|  File Storage Blocks fileBlocks (fld)

typedef struct _LFS_LLFSROOTLIST_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* File Dump Flags */
    LFS_DUMPSTFLAGS      dmpFlags;
```

```
    /* File Storage Flags */
    LFS_FILESTFLAGS      fileFlags;

    /* Byte pad */
    char                 byte_pad0[2];

    /* File Storage ID */
    LFS_FILESTID         fileId;

    /* File Storage Update Time */
    unsigned int         fileTime;

    /* File Storage Size */
    unsigned int         fileSize;

    /* File Storage Blocks */
    unsigned int         fileBlocks;

} LFS_LLFSROOTLIST_Tlm;
```

Fields:

`DUMPSTFLAGS` **(9.4.0)**

Attribute(s):

Brief: "File Dump Flags"

Instance(s): `dmpFlags`

`FILESTBLOCK` **(9.5.7)**

Attribute(s):

Brief: "File Storage Blocks"

Instance(s): `fileBlocks`

`FILESTFLAGS` **(9.4.1)**

Attribute(s):

Brief: "File Storage Flags"

Instance(s): `fileFlags`

`FILESTID` **(9.4.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`FILESTSIZE` **(9.5.11)**

Attribute(s):

Brief: "File Storage Size"

Instance(s): `fileSize`

`FILESTTIME` **(9.5.12)**

Attribute(s):

Brief: "File Storage Update Time"

Instance(s): `fileTime`

9.3.3 `LLFSSYSLIST` (794/0x31A)

Context:

Description:

"File System Status Report" Telemetry Packet

Provides the status of a LAT file system partition. One of these packets is sent in response to the LLFSSYSSTATUS telecommand.

Layout:

File System Status Report (LLFSSYSLIST) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x31A**										
002	**SF**		**Sequence Count**													
004	**Packet Length=144**															
006	**Timestamp**															
00E	**DUMPSTFLAGS:** `dmpFlags`															
010	**FILESTID:** `fileId`															
014	**SYSBLKSIZE:** `blkSize`															
018	**SYSBLKFREE:** `blkFree`															
01C	**SYSBLKTOTAL:** `blkTotal`															

```
LLFSSYSLIST (pkt)
|  File Dump Flags dmpFlags (bf)
```

```
|   |  LAT Storage Unit latUnit (0-3)
|   |  Transaction ID xactId (4-15)
|  File Storage ID fileId (bf)
|   |  File ID Device Number dev (0-2)
|   |  File ID Directory Number dir (3-9)
|   |  File ID File Number file (10-31)
|  File System Block Size blkSize (fld)
|  File System Free Blocks blkFree (fld)
|  File System Total Blocks blkTotal (fld)

typedef struct _LFS_LLFSSYSLIST_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* File Dump Flags */
    LFS_DUMPSTFLAGS      dmpFlags;

    /* File Storage ID */
    LFS_FILESTID         fileId;

    /* File System Block Size */
    unsigned int         blkSize;

    /* File System Free Blocks */
    unsigned int         blkFree;

    /* File System Total Blocks */
    unsigned int         blkTotal;

} LFS_LLFSSYSLIST_Tlm;
```

Fields:

`DUMPSTFLAGS` **(9.4.0)**

Attribute(s):

Brief: "File Dump Flags"

Instance(s): `dmpFlags`

`FILESTID` **(9.4.2)**

Attribute(s):

Brief: "File Storage ID"

Instance(s): `fileId`

`SYSBLKFREE` **(9.5.15)**

Attribute(s):

Brief: "File System Free Blocks"

Instance(s): `blkFree`

`SYSBLKSIZE` **(9.5.16)**

Attribute(s):

Brief: "File System Block Size"

Instance(s): `blkSize`

`SYSBLKTOTAL` **(9.5.17)**

Attribute(s):

Brief: "File System Total Blocks"

Instance(s): `blkTotal`

9.4 Telemetry Bitfields

9.4.0 `DUMPSTFLAGS` (File Dump Flags) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**3**	**4**	`LATSTUNIT latUnit`
4	**15**	**12**	`LFSXID xactId`

```
DUMPSTFLAGS (bf)
|  LAT Storage Unit latUnit (0-3)
|  Transaction ID xactId (4-15)
```

```
typedef struct _LFS_DUMPSTFLAGS_Bfl {

    #if ENDIANNESS_IS_BIG

    /* LAT Storage Unit */
    unsigned short       latUnit:4;

    /* Transaction ID */
    unsigned short       xactId:12;

    #elif ENDIANNESS_IS_LITTLE

    /* Transaction ID */
    unsigned short       xactId:12;

    /* LAT Storage Unit */
    unsigned short       latUnit:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LFS_DUMPSTFLAGS_Bfl;

typedef union _LFS_DUMPSTFLAGS {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LFS_DUMPSTFLAGS_Bfl     bf;
```

```
} LFS_DUMPSTFLAGS;
```

Fields:

`LATSTUNIT` **(9.5.13)**

Attribute(s):

Brief: "LAT Storage Unit"

Instance(s): `latUnit`

`LFSXID` **(9.5.14)**

Attribute(s):

Brief: "Transaction ID"

Instance(s): `xactId`

Used by:

`LLFSDIRLIST (9.3.0)`

9.4.1 **`FILESTFLAGS`** (File Storage Flags) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**0**	**1**	`FILESTARCHIVE archiveFlag`
1	**1**	**1**	`FILESTDIR dirFlag`
2	**2**	**1**	`FILESTRDONLY roFlag`

```
FILESTFLAGS (bf)
|  File Storage Archive Flag archiveFlag (0)
|  File Storage Directory Flag dirFlag (1)
|  File Storage Read-Only Flag roFlag (2)

typedef struct _LFS_FILESTFLAGS_Bfl {

    #if ENDIANNESS_IS_BIG

    /* File Storage Archive Flag */
    unsigned short       archiveFlag:1;

    /* File Storage Directory Flag */
```

```
    unsigned short       dirFlag:1;

    /* File Storage Read-Only Flag */
    unsigned short       roFlag:1;

    /* Bit pad */
    unsigned short       bit_pad:13;

    #elif ENDIANNESS_IS_LITTLE

    /* Bit pad */
    unsigned short       bit_pad:13;

    /* File Storage Read-Only Flag */
    unsigned short       roFlag:1;

    /* File Storage Directory Flag */
    unsigned short       dirFlag:1;

    /* File Storage Archive Flag */
    unsigned short       archiveFlag:1;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LFS_FILESTFLAGS_Bfl;

typedef union _LFS_FILESTFLAGS {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LFS_FILESTFLAGS_Bfl     bf;

} LFS_FILESTFLAGS;
```

Fields:

`FILESTARCHIVE` **(9.5.6)**

Attribute(s):

Brief: "File Storage Archive Flag"

Instance(s): `archiveFlag`

`FILESTDIR` **(9.5.8)**

Attribute(s):

Brief: "File Storage Directory Flag"

Instance(s): `dirFlag`

`FILESTRDONLY` **(9.5.10)**

Attribute(s):

Brief: "File Storage Read-Only Flag"

Instance(s): `roFlag`

Used by:

`LLFSDIRLIST (9.3.0)`

9.4.2 `FILESTID` (File Storage ID) Telemetry Bitfield

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**2**	**3**	`FILEDEV dev`
3	**9**	**7**	`FILEDIR dir`
10	**31**	**22**	`FILENUM file`

```
FILESTID (bf)
|  File ID Device Number dev (0-2)
|  File ID Directory Number dir (3-9)
|  File ID File Number file (10-31)
```

```
typedef struct _LFS_FILESTID_Bfl {

    #if ENDIANNESS_IS_BIG

    /* File ID Device Number */
    unsigned int         dev:3;

    /* File ID Directory Number */
    unsigned int         dir:7;

    /* File ID File Number */
    unsigned int         file:22;

    #elif ENDIANNESS_IS_LITTLE
```

```
        /* File ID File Number */
        unsigned int         file:22;

        /* File ID Directory Number */
        unsigned int         dir:7;

        /* File ID Device Number */
        unsigned int         dev:3;

        #else

        #error _FILE_ ENDIANNESS NOT DEFINED

        #endif

    } __attribute__((packed)) LFS_FILESTID_Bfl;

    typedef union _LFS_FILESTID {

        /* underlying unsigned int */
        unsigned int                     ui;

        /* struct mapped bitfield */
        struct _LFS_FILESTID_Bfl         bf;

    } LFS_FILESTID;
```

Fields:

`FILEDEV` **(9.5.1)**

Attribute(s):

Brief: "File ID Device Number"

Instance(s): `dev`

`FILEDIR` **(9.5.2)**

Attribute(s):

Brief: "File ID Directory Number"

Instance(s): `dir`

`FILENUM` **(9.5.5)**

Attribute(s):

Brief: "File ID File Number"

Instance(s): `file`

Used by:

`LLFSDIRLIST (9.3.0)`

9.5 Telemetry Fields

9.5.0 `FILEDATA` (File Data Contents) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`LLFSDUMPCTDB (9.3.1)`

9.5.1 `FILEDEV` (File ID Device Number) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	3 bits

Description:

Used by:

`FILESTID (9.4.2)`

9.5.2 `FILEDIR` (File ID Directory Number) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	7 bits

Description:

Used by:

`FILESTID (9.4.2)`

9.5.3 `FILEDUMPSIZE` (File Dump Current Size) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDUMPCTDB (9.3.1)`

9.5.4 `FILEHDR` (File Header Data) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`LLFSDIRLIST (9.3.0)`

9.5.5 `FILENUM` (File ID File Number) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	22 bits

Description:

Used by:

`FILESTID (9.4.2)`

9.5.6 `FILESTARCHIVE` (File Storage Archive Flag) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Used by:

`FILESTFLAGS (9.4.1)`

9.5.7 `FILESTBLOCK` (File Storage Blocks) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDIRLIST (9.3.0)`

9.5.8 `FILESTDIR` (File Storage Directory Flag) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	1 bit

Description:

Used by:

`FILESTFLAGS (9.4.1)`

9.5.9 `FILESTOFFSET` (File Data Offset) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDUMPCTDB (9.3.1)`

9.5.10 `FILESTRDONLY` (File Storage Read-Only Flag) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	1 bit

Description:

Used by:

`FILESTFLAGS (9.4.1)`

9.5.11 `FILESTSIZE` (File Storage Size) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDIRLIST (9.3.0)`

9.5.12 `FILESTTIME` (File Storage Update Time) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSDIRLIST (9.3.0)`

9.5.13 `LATSTUNIT` (LAT Storage Unit) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	4 bits

Description:

Used by:

`DUMPSTFLAGS (9.4.0)`

9.5.14 `LFSXID` (Transaction ID) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	12 bits

Description:

Used by:

`DUMPSTFLAGS (9.4.0)`

9.5.15 `SYSBLKFREE` (File System Free Blocks) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSSYSLIST (9.3.3)`

9.5.16 `SYSBLKSIZE` (File System Block Size) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSSYSLIST (9.3.3)`

9.5.17 `SYSBLKTOTAL` (File System Total Blocks) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLFSSYSLIST (9.3.3)`

10 LHK Package

10.0 Overview

The LHK package provides monitoring facilities that are orthogonal and asynchronous to the monitoring that is based on the contents of physics events being read from the instrument. This is much more fundamental monitoring (e.g., temperatures, voltages, currents). LHK runs its own task.

The package supports the following functions:

- Housekeeping data stream

10.1 Command Packets

ITOS Mnemonic: `LHKREQDIAGPKT`

10.1.0 `ReqDiagPacket` (1616/0x650:0)

Context:

Description:

"Request a Housekeeping Diagnostic Packet" Telecommand Packet

Sends a request to the housekeeping system to send a packets using the diagnostic channel.

Layout:

Request a Housekeeping Diagnostic Packet (ReqDiagPacket) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x650:0**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=15**															
006	**0**	**Function Code=0**														
008	**APID:** apid															
00C	**DiagPktCnt:** count															
010	**DiagInterval:** interval															
014	**Packet Checksum**															

```
ReqDiagPacket (pkt)
|  Application ID parameter apid (fld)
|  Diagnostic sample Packet Count count (fld)
|  Diagnostic sample interval interval (fld)

typedef struct _LHK_ReqDiagPacket_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LHK_ReqDiagPacket_Prm pay;
```

```
} LHK_ReqDiagPacket_Pkt;
```

Fields:

`APID` **(10.2.0)**

Attribute(s):
rng: LHKAPIDRNG (10.3.0)

Brief: "Application ID parameter"

Instance(s): `apid`

`DiagInterval` **(10.2.1)**

Attribute(s):
rng: LHKDIAGINTV (10.3.1)

Brief: "Diagnostic sample interval"

Instance(s): `interval`

Specifies interval in milliseconds between diagnostic packet samples.

`DiagPktCnt` **(10.2.2)**

Attribute(s):
rng: LHKDIAGPKTCNT (10.3.2)

Brief: "Diagnostic sample Packet Count"

Instance(s): `count`

Specifies the number of diagnostic packets to sample.

ITOS Mnemonic: `LHKSTOPDIAG`

10.1.1 `StopDiag` (1616/0x650:2)

Context:

Description:

"Stop Diagnostic Sample" Telecommand Packet

Terminates the current diagnostic sample run.

Layout:

Stop Diagnostic Sample (StopDiag) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x650:2										
002	SF=3		Sequence Count													
004	Packet Length=3															
006	0	Function Code=2														

```
StopDiag (pkt)

typedef struct _LHK_StopDiag_Pkt {

    /* CCSDS header */
    char                 hdr[8];

} LHK_StopDiag_Pkt;
```

ITOS Mnemonic: `LHKSYSRESET`

10.1.2 `SysReset` (1616/0x650:1)

Context:

Description:

"System Reset" Telecommand Packet

Resets the housekeeping system using configuration file parameters. File IDs can be null, which specifies loading of default configuration.

Layout:

System Reset (SysReset) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x650:1										
002	SF=3		Sequence Count													
004	Packet Length=11															
006	0	Function Code=1														
008	FileId: file0															
00C	FileId: file1															
010	Packet Checksum															

```
SysReset (pkt)
|  Configuration File ID file0 (fld) ... file1 (fld)

typedef struct _LHK_SysReset_Pkt {

    /* CCSDS header */
```

```
      char                 hdr[8];

      /* Command payload */
      LHK_SysReset_Prm     pay;

  } LHK_SysReset_Pkt;
```

Fields:

`FileId` **(10.2.3)**

Attribute(s):

Brief: "Configuration File ID"

Instance(s): `file0`, `file1`

32 bit value specifying the configuration file ID, which conforms to the FILE standard convention.

10.2 Command Fields

10.2.0 `APID` (Application ID parameter) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`ReqDiagPacket (10.1.0)`

10.2.1 `DiagInterval` (Diagnostic sample interval) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Specifies interval in milliseconds between diagnostic packet samples.

Used by:

`ReqDiagPacket (10.1.0)`

10.2.2 `DiagPktCnt` (Diagnostic sample Packet Count) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Specifies the number of diagnostic packets to sample.

Used by:

`ReqDiagPacket (10.1.0)`

10.2.3 `FileId` (Configuration File ID) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

32 bit value specifying the configuration file ID, which conforms to the FILE standard convention.

Used by:

`SysReset (10.1.2)`

10.3 Ranges

10.3.0 `LHKAPIDRNG` (LHK APID Range) Range

ITOS Mnemonic: `LHKAPIDRNG`

Description:

Valid range of APIDs for LHK

Definition:

Limits 528 - 555

Used by:

`ReqDiagPacket (10.1.0)`

10.3.1 `LHKDIAGINTV` (Diagnostic Interval) Range

ITOS Mnemonic: `LHKDIAGINTV`

Description:

Range in milliseconds for a diagnostic sample interval.

Definition:

Limits 100 - 4000

Used by:

`ReqDiagPacket (10.1.0)`

10.3.2 `LHKDIAGPKTCNT` (Diagnostic Packet Count) Range

ITOS Mnemonic: `LHKDIAGPKTCNT`

Description:

Specifies the range of valid diagnostic sample packet counts.

Definition:

Limits 1 - 64

Used by:

`ReqDiagPacket (10.1.0)`

10.4 Telemetry Packets

10.4.0 `AemEnv0` (550/0x226)

Context:

Description:

"AEM Environmental Monitor Packet 0" Telemetry Packet

AEM Packet 0

Layout:

AEM Environmental Monitor Packet 0 (`AemEnv0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x226**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-3]`															
012	**AEMFRPWRREG:** `pwrst`															
014	**AEMFRENV:** `frees[0-11]`															

```
AemEnv0 (pkt)
|  Spare byte field spares[0-3] (fld)
|  AEM FREE board power status register pwrst (bf)
|  |  Spare 4 bits spare (0-3)
|  |  AEM FREE Board Power State free11 (4) ... free0 (15)
|  AEM Free Board Environmental ADCs frees[0-11] (str)
|  |  AEM VDD vdd (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  AEM Free board Temperature temp (same bf)
|  |  AEM Free Board HV1 hv1 (same bf)
|  |  AEM Free Board HV2 hv2 (same bf)
```

```
typedef struct _LHK_AemEnv0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[4];

    /* AEM FREE board power status register */
    LHK_AEMFRPWRREG      pwrst;

    /* AEM Free Board Environmental ADCs */
    LHK_AEMFRENV         frees[12];
```

```
} LHK_AemEnv0_Tlm;
```

Fields:

`AEMFRENV` **(10.5.1)**

Attribute(s):

Brief: "AEM Free Board Environmental ADCs"

Instance(s): `frees`

`AEMFRPWRREG` **(10.6.8)**

Attribute(s):

Brief: "AEM FREE board power status register"

Instance(s): `pwrst`

Contains the status of the power supplied by the AEM to its FREE boards.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.1 `CmdCnt0` (552/0x228)

Context:

Description:

"Command Statistics Packet 0" Telemetry Packet

Contains task level command statistics.

Layout:

Command Statistics Packet 0 (`CmdCnt0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x228**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															

Command Statistics Packet 0 (`CmdCnt0`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
006	**Timestamp**															
00E	**TSP16:** `spares[0-2]`															
014	**CMDCNTRS:** `cntrs[0-5]`															

```
CmdCnt0 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  Command Counters cntrs[0-5] (str)
|  |  Timestamp seconds sec (fld)
|  |  Timestamp subseconds subsec (fld)
|  |  Commands sent counter sent (fld)
|  |  Command dispatch failure counter dispf (fld)
|  |  Command execution failure counter exef (fld)
```

```
typedef struct _LHK_CmdCnt0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* Command Counters */
    LHK_CMDCNTRS         cntrs[6];

} LHK_CmdCnt0_Tlm;
```

Fields:

`CMDCNTRS` **(10.5.2)**

Attribute(s):

Brief: "Command Counters"

Instance(s): `cntrs`

Structure to hold per task command count information.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.2 `CmdCnt1` (553/0x229)

Context:

Description:

"Command Statistics Packet 1" Telemetry Packet

Contains task level command statistics

Layout:

Command Statistics Packet 1 (CmdCnt1) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x229**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** `spares[0-2]`															
014	**CMDCNTRS:** `cntrs[0-5]`															

```
CmdCnt1 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  Command Counters cntrs[0-5] (str)
|  |  Timestamp seconds sec (fld)
|  |  Timestamp subseconds subsec (fld)
|  |  Commands sent counter sent (fld)
|  |  Command dispatch failure counter dispf (fld)
|  |  Command execution failure counter exef (fld)

typedef struct _LHK_CmdCnt1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* Command Counters */
    LHK_CMDCNTRS         cntrs[6];

} LHK_CmdCnt1_Tlm;
```

Fields:

`CMDCNTRS` **(10.5.2)**

Attribute(s):

Brief: "Command Counters"

Instance(s): `cntrs`

Structure to hold per task command count information.

`TSP16` **(10.7.54)**

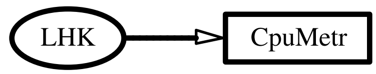
Attribute(s):

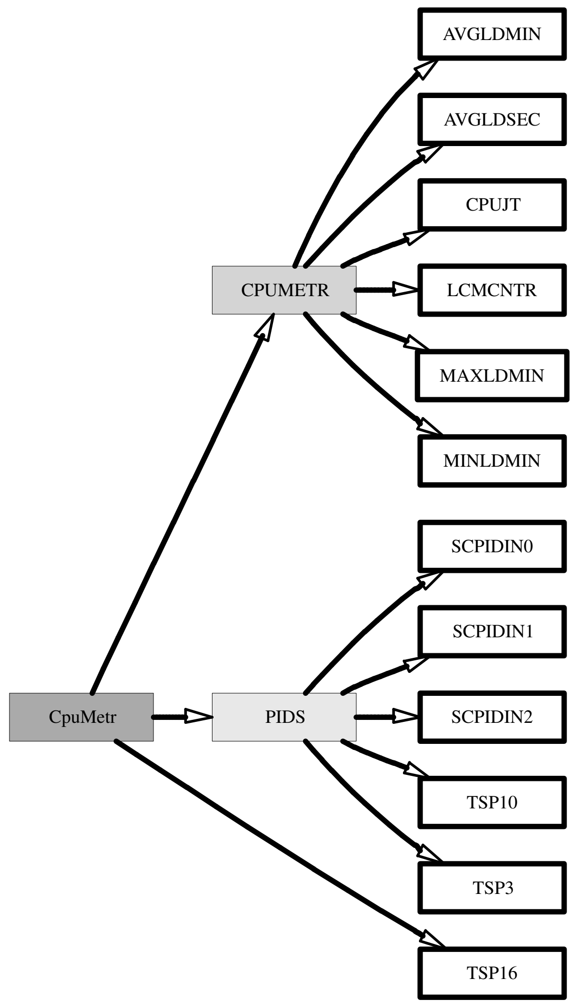
Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.3 `CpuMetr` (555/0x22B)

Context:





Description:

"CPU Metrics/RT Statistics Packet" Telemetry Packet

Contains CPU metrics for all LAT CPUs. Contains the 1553 RT driver statistics.

Layout:

CPU Metrics/RT Statistics Packet (CpuMetr) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x22B**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spares[0-2]															
014	**RTSTATS:** rtstat															
03C	**CPUMETR:** cpum[0-3]															
044	**TSP16:** pad[0-23]															

```
CpuMetr (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  1553 Remote Terminal Statistics rtstat (str)
|  |  1553 Error Count err (fld)
|  |  1553 Interrupts intr (fld)
|  |  1553 Cmd Rx Packet Count cxp (fld)
|  |  1553 Cmd Rx Byte Count cxb (fld)
|  |  1553 Cmd Tx Packet Count txp (fld)
|  |  1553 Cmd Tx Byte Count txb (fld)
|  |  1553 Housekeeping Packet Count hkp (fld)
|  |  1553 Housekeeping Byte Count hkb (fld)
|  |  1553 Telemetry Packet Count tlmp (fld)
|  |  1553 Telemertry Byte Count tlmb (fld)
|  CPU Metrics cpum[0-3] (str)
|  |  CPU Junction Temperature temp (fld)
|  Spare 16 bit field pad[0-23] (fld)
```

```
typedef struct _LHK_CpuMetr_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* 1553 Remote Terminal Statistics */
    LHK_RTSTATS          rtstat;

    /* CPU Metrics */
    LHK_CPUMETR          cpum[4];

    /* Spare 16 bit field */
    unsigned short       pad[24];

} LHK_CpuMetr_Tlm;
```

Fields:

CPUMETR **(10.5.3)**

Attribute(s):

Brief: "CPU Metrics"

Instance(s): `cpum`

Structure containing metrics for a single CPU.

`RTSTATS` **(10.5.7)**

Attribute(s):

Brief: "1553 Remote Terminal Statistics"

Instance(s): `rtstat`

Contains the 1553 RT driver statistics.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `pad`, `spares`

10.4.4 `DiagAemEnv0` (646/0x286)

Context:

Description:

"Diagnostic AEM Environmental Monitor Packet 0" Telemetry Packet

AEM Packet 0

Layout:

Diagnostic AEM Environmental Monitor Packet 0 (`DiagAemEnv0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x286**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-3]`															
012	**AEMFRPWRREG:** `pwrst`															

Diagnostic AEM Environmental Monitor Packet 0 (DiagAemEnv0) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
014	**AEMFRENV:** frees[0-11]															

```
DiagAemEnv0 (pkt)
|  Spare byte field spares[0-3] (fld)
|  AEM FREE board power status register pwrst (bf)
|  |  Spare 4 bits spare (0-3)
|  |  AEM FREE Board Power State free11 (4) ... free0 (15)
|  AEM Free Board Environmental ADCs frees[0-11] (str)
|  |  AEM VDD vdd (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  AEM Free board Temperature temp (same bf)
|  |  AEM Free Board HV1 hv1 (same bf)
|  |  AEM Free Board HV2 hv2 (same bf)
```

```
typedef struct _LHK_DiagAemEnv0_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[4];

    /* AEM FREE board power status register */
    LHK_AEMFRPWRREG     pwrst;

    /* AEM Free Board Environmental ADCs */
    LHK_AEMFRENV        frees[12];

} LHK_DiagAemEnv0_Tlm;
```

Fields:

AEMFRENV **(10.5.1)**

Attribute(s):

Brief: "AEM Free Board Environmental ADCs"

Instance(s): frees

AEMFRPWRREG **(10.6.8)**

Attribute(s):

Brief: "AEM FREE board power status register"

Instance(s): `pwrst`

Contains the status of the power supplied by the AEM to its FREE boards.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.5 `DiagCmdCnt0` (648/0x288)

Context:

Description:

"Diagnostic Command Statistics Packet 0" Telemetry Packet

Contains task level command statistics

Layout:

Diagnostic Command Statistics Packet 0 (`DiagCmdCnt0`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x288**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** `spares[0-2]`															
014	**CMDCNTRS:** `cntrs[0-5]`															

```
DiagCmdCnt0 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  Command Counters cntrs[0-5] (str)
|  |  Timestamp seconds sec (fld)
|  |  Timestamp subseconds subsec (fld)
|  |  Commands sent counter sent (fld)
|  |  Command dispatch failure counter dispf (fld)
|  |  Command execution failure counter exef (fld)

typedef struct _LHK_DiagCmdCnt0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];
```

```
    /* Command Counters */
    LHK_CMDCNTRS         cntrs[6];

} LHK_DiagCmdCnt0_Tlm;
```

Fields:

`CMDCNTRS` **(10.5.2)**

Attribute(s):

Brief: "Command Counters"

Instance(s): `cntrs`

Structure to hold per task command count information.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.6 `DiagCmdCnt1` (649/0x289)

Context:

Description:

"Diagnostic Command Statistics Packet 1" Telemetry Packet

Contains task level command statistics.

Layout:

Diagnostic Command Statistics Packet 1 (DiagCmdCnt1) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x289**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spares[0-2]															
014	**CMDCNTRS:** cntrs[0-5]															

```
DiagCmdCnt1 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
```

```
|  Command Counters cntrs[0-5] (str)
|  |  Timestamp seconds sec (fld)
|  |  Timestamp subseconds subsec (fld)
|  |  Commands sent counter sent (fld)
|  |  Command dispatch failure counter dispf (fld)
|  |  Command execution failure counter exef (fld)

typedef struct _LHK_DiagCmdCnt1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* Command Counters */
    LHK_CMDCNTRS         cntrs[6];

} LHK_DiagCmdCnt1_Tlm;
```

Fields:

`CMDCNTRS` **(10.5.2)**

Attribute(s):

Brief: "Command Counters"

Instance(s): `cntrs`

Structure to hold per task command count information.

`TSP16` **(10.7.54)**

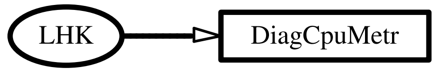
Attribute(s):

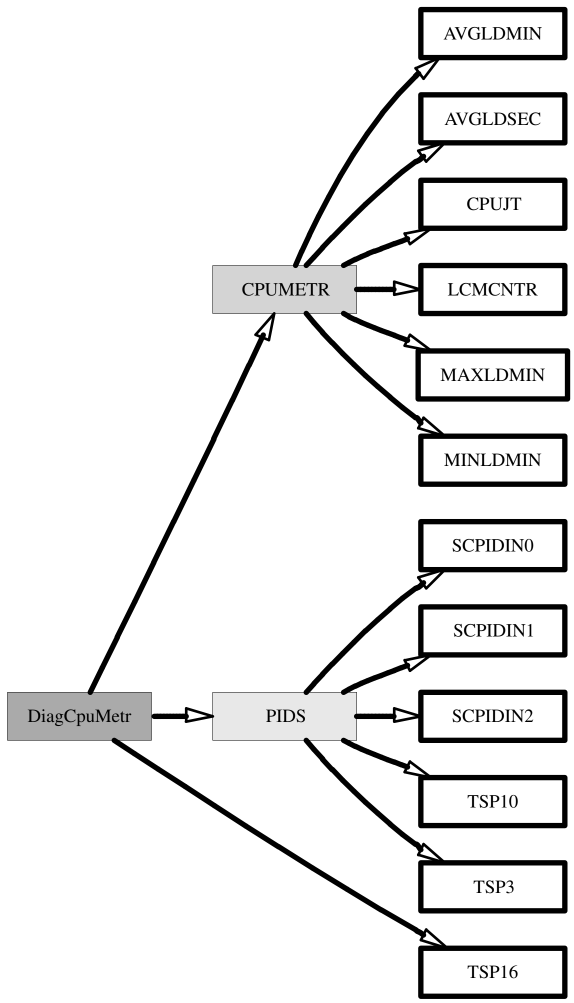
Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.7 `DiagCpuMetr` (651/0x28B)

Context:





Description:

"Diagnostic CPU Metrics/RT Statistics Packet" Telemetry Packet

Contains CPU metrics for all LAT CPUs. Contains the 1553 RT driver statistics.

Layout:

<table>
<tr><th colspan="17">Diagnostic CPU Metrics/RT Statistics Packet (DiagCpuMetr)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x28B</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=109</td></tr>
<tr><td>006</td><td colspan="16">Timestamp</td></tr>
<tr><td>00E</td><td colspan="16">TSP16: spares[0-2]</td></tr>
<tr><td>014</td><td colspan="16">RTSTATS: rtstat</td></tr>
<tr><td>03C</td><td colspan="16">CPUMETR: cpum[0-3]</td></tr>
<tr><td>044</td><td colspan="16">TSP16: pad[0-23]</td></tr>
</table>

```
DiagCpuMetr (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  1553 Remote Terminal Statistics rtstat (str)
|  |  1553 Error Count err (fld)
|  |  1553 Interrupts intr (fld)
|  |  1553 Cmd Rx Packet Count cxp (fld)
|  |  1553 Cmd Rx Byte Count cxb (fld)
|  |  1553 Cmd Tx Packet Count txp (fld)
|  |  1553 Cmd Tx Byte Count txb (fld)
|  |  1553 Housekeeping Packet Count hkp (fld)
|  |  1553 Housekeeping Byte Count hkb (fld)
|  |  1553 Telemetry Packet Count tlmp (fld)
|  |  1553 Telemertry Byte Count tlmb (fld)
|  CPU Metrics cpum[0-3] (str)
|  |  CPU Junction Temperature temp (fld)
|  Spare 16 bit field pad[0-23] (fld)
```

```
typedef struct _LHK_DiagCpuMetr_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* 1553 Remote Terminal Statistics */
    LHK_RTSTATS          rtstat;

    /* CPU Metrics */
    LHK_CPUMETR          cpum[4];

    /* Spare 16 bit field */
    unsigned short       pad[24];

} LHK_DiagCpuMetr_Tlm;
```

Fields:

CPUMETR **(10.5.3)**

Attribute(s):

Brief: "CPU Metrics"

Instance(s): `cpum`

Structure containing metrics for a single CPU.

`RTSTATS` **(10.5.7)**

Attribute(s):

Brief: "1553 Remote Terminal Statistics"

Instance(s): `rtstat`

Contains the 1553 RT driver statistics.

`TSP16` **(10.7.54)**

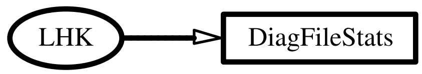
Attribute(s):

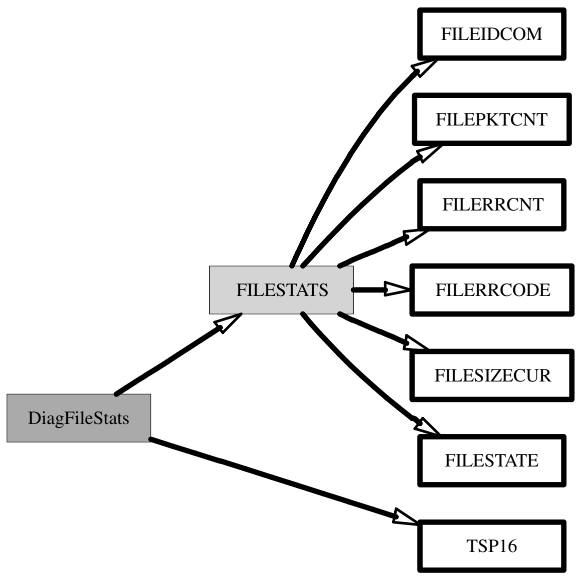
Brief: "Spare 16 bit field"

Instance(s): `pad`, `spares`

10.4.8 `DiagFileStats` (650/0x28A)

Context:





Description:

"Diagnositc File System Statistics Packet" Telemetry Packet

Contains FILE system statistics for each CPU.

Layout:

Diagnositc File System Statistics Packet `(DiagFileStats)` **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x28A**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** `spares[0-2]`															
014	**FILESTATS:** `siu`															
02C	**FILESTATS:** `epu0`															
044	**FILESTATS:** `epu1`															
05C	**FILESTATS:** `epu2`															

```
DiagFileStats (pkt)
|  Spare 16 bit field spares[0-2] (fld)
```

```
|  FILE Upload Statistics siu (str)
|  |  FILE Upload State state (fld)
|  |  File Size Current cursize (fld)
|  |  FILE Packet Count pktcnt (fld)
|  |  FILE Error Code errcode (fld)
|  |  FILE Error Count errcnt (fld)
|  |  FILE ID Commit comid (fld)
|  FILE Upload Statistics epu0 (same str)
|  FILE Upload Statistics epu1 (same str)
|  FILE Upload Statistics epu2 (same str)
```

```
typedef struct _LHK_DiagFileStats_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* FILE Upload Statistics */
    LHK_FILESTATS        siu;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu0;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu1;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu2;

} LHK_DiagFileStats_Tlm;
```

Fields:

`FILESTATS` **(10.5.4)**

Attribute(s):

Brief: "FILE Upload Statistics"

Instance(s): `epu0`, `epu1`, `epu2`, `siu`

Describes the FILE upload current state.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.9 `DiagLrs0` (647/0x287)

Context:

Description:

"Diagnostic Low-rate Science Packet" Telemetry Packet

Contains TEM dead-time counters and GEM livetime counter

Layout:

Diagnostic Low-rate Science Packet (DiagLrs0) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x287**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spares[0-2]															
014	**TIMESEC:** gemsec															
018	**TIMESUBSEC:** gemsub															
01C	**GEMLRS:** gemlrs															
02C	**TIMESEC:** temsec															
030	**TIMESUBSEC:** temsub															
034	**TEMDEADTIMEREG:** temlrs[0-15]															

```
DiagLrs0 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  Timestamp seconds gemsec (fld)
|  Timestamp subseconds gemsub (fld)
|  GEM Low-Rate Science Counter gemlrs (str)
|  |  GEM low-rate science livetime counter live (fld)
|  |  Low-rate science prescaled counter presc (fld)
|  |  GEM low-rate science discarded counter disc (fld)
|  |  GEM low-rate science sent counter sent (fld)
|  Timestamp seconds temsec (fld)
|  Timestamp subseconds temsub (fld)
|  TEM Deadtime Register Bitfield temlrs[0-15] (bf)
|  |  TEM Deadtime Low-rate Science Counter deadtime (0-23)

typedef struct _LHK_DiagLrs0_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare 16 bit field */
    unsigned short      spares[3];
```

```
    /* Timestamp seconds */
    unsigned int         gemsec;

    /* Timestamp subseconds */
    unsigned int         gemsub;

    /* GEM Low-Rate Science Counter */
    LHK_GEMLRS           gemlrs;

    /* Timestamp seconds */
    unsigned int         temsec;

    /* Timestamp subseconds */
    unsigned int         temsub;

    /* TEM Deadtime Register Bitfield */
    LHK_TEMDEADTIMEREG   temlrs[16];

} LHK_DiagLrs0_Tlm;
```

Fields:

`GEMLRS` **(10.5.5)**

Attribute(s):

Brief: "GEM Low-Rate Science Counter"

Instance(s): `gemlrs`

Contains 4 GEM low-rate science counter values.

`TEMDEADTIMEREG` **(10.6.29)**

Attribute(s):

Brief: "TEM Deadtime Register Bitfield"

Instance(s): `temlrs`

Describes the TEM dead time low-rate science counter register.

`TIMESEC` **(10.7.48)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `gemsec`, `temsec`

`TIMESUBSEC` **(10.7.49)**

Attribute(s):

Brief: "Timestamp subseconds"

Instance(s): `gemsub`, `temsub`

`TSP16` **(10.7.54)**

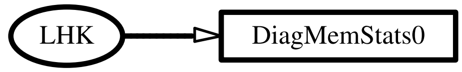
Attribute(s):

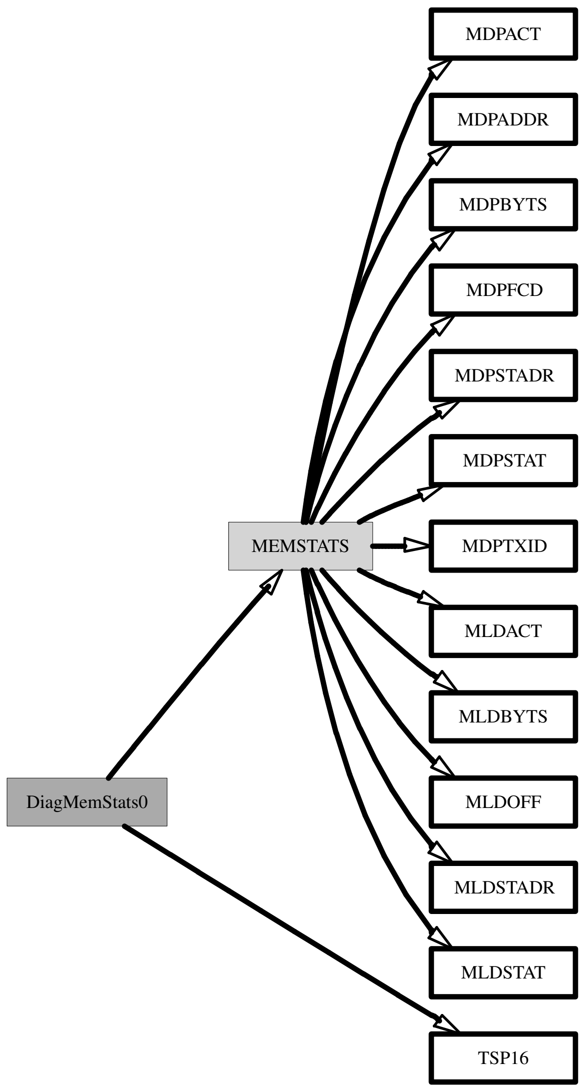
Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.10 `DiagMemStats0` (652/0x28C)

Context:





Description:

"Diagnostic Memory Load/Dump Statistics" Telemetry Packet

Contains the statistics for memory loads and dumps on the SIU and EPU0.

Layout:

Diagnostic Memory Load/Dump Statistics (`DiagMemStats0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x28C**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** `spare[0-2]`															
014	**MEMSTATS:** `mem[0-1]`															

```
DiagMemStats0 (pkt)
|  Spare 16 bit field spare[0-2] (fld)
|  MEM Statistics mem[0-1] (str)
|  |  Status of most recent load action ldstat (fld)
|  |  Memory load active flag ldact (fld)
|  |  Starting memory load address ldstart (fld)
|  |  Memory load total bytes ldbytes (fld)
|  |  Memory load offset ldoffset (fld)
|  |  Memory dump status dmpstat (fld)
|  |  Memory dump active dmpact (fld)
|  |  Memory dump start address dmpstart (fld)
|  |  Memory dump bytes dmpbytes (fld)
|  |  Memory dump address dmpaddr (fld)
|  |  Memory dump function code dmpfcode (fld)
|  |  Memory dump transaction ID dmptid (fld)
```

```
typedef struct _LHK_DiagMemStats0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spare[3];

    /* MEM Statistics */
    LHK_MEMSTATS         mem[2];

} LHK_DiagMemStats0_Tlm;
```

Fields:

`MEMSTATS` **(10.5.6)**

Attribute(s):

Brief: "MEM Statistics"

Instance(s): `mem`

Contains statistics for the MEM task.

`TSP16` **(10.7.54)**

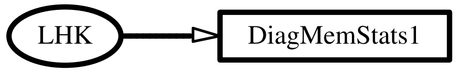
Attribute(s):

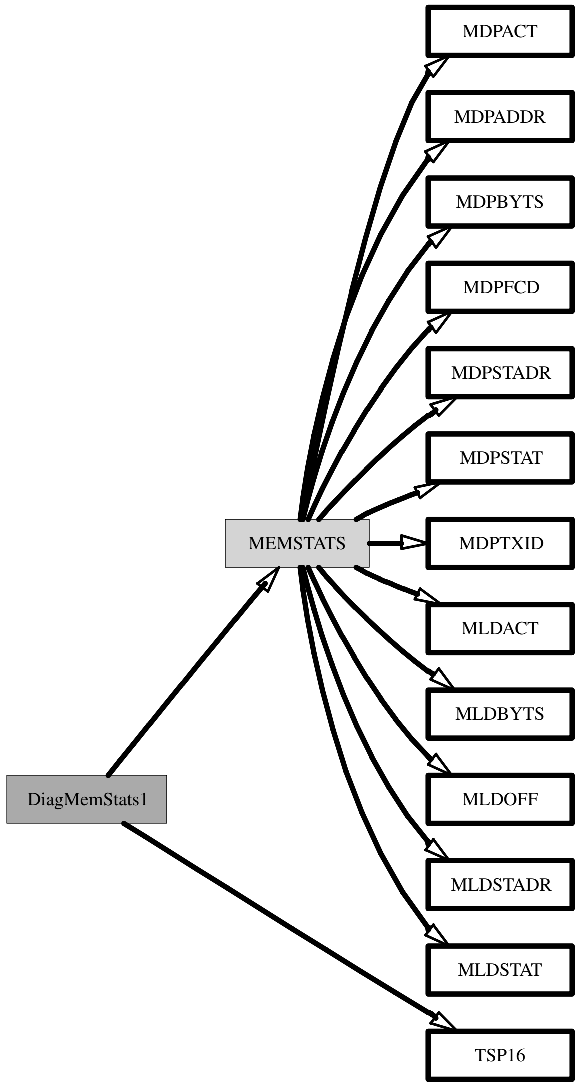
Brief: "Spare 16 bit field"

Instance(s): `spare`

10.4.11 `DiagMemStats1` (653/0x28D)

Context:





Description:

"Diagnostic Memory Load/Dump Statistics" Telemetry Packet

Contains the statistics for memory loads and dumps on the EPU1 and EPU2.

Layout:

Diagnostic Memory Load/Dump Statistics (DiagMemStats1) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x28D										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP16:** spares[0-2]															
014	**MEMSTATS:** mem[0-1]															

```
DiagMemStats1 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  MEM Statistics mem[0-1] (str)
|  |  Status of most recent load action ldstat (fld)
|  |  Memory load active flag ldact (fld)
|  |  Starting memory load address ldstart (fld)
|  |  Memory load total bytes ldbytes (fld)
|  |  Memory load offset ldoffset (fld)
|  |  Memory dump status dmpstat (fld)
|  |  Memory dump active dmpact (fld)
|  |  Memory dump start address dmpstart (fld)
|  |  Memory dump bytes dmpbytes (fld)
|  |  Memory dump address dmpaddr (fld)
|  |  Memory dump function code dmpfcode (fld)
|  |  Memory dump transaction ID dmptid (fld)
```

```
typedef struct _LHK_DiagMemStats1_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare 16 bit field */
    unsigned short      spares[3];

    /* MEM Statistics */
    LHK_MEMSTATS        mem[2];

} LHK_DiagMemStats1_Tlm;
```

Fields:

MEMSTATS **(10.5.6)**

Attribute(s):

Brief: "MEM Statistics"

Instance(s): mem

Contains statistics for the MEM task.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.12 `DiagPduEnv0` (638/0x27E)

Context:

Description:

"Diagnostic PDU Environmental Packet 0" Telemetry Packet

PDU Packet 0

Layout:

Diagnostic PDU Environmental Packet 0 (DiagPduEnv0) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x27E**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**PDUTEMPWRREG:** `tempwr`															
016	**PDUEPUCRATEPWR:** `epupwr`															
018	**ACDPWRREG:** `acdpwr`															
01A	**TEM33V:** `temvolt[0-15]`															
03A	**TEMPSTEMP:** `temps[0-15]`															
05A	**EPUV:** `epuvolt[0-2]`															
060	**EPUTEMP:** `eputemp[0-2]`															
066	**AEMDAQENV:** `dab`															
06E	**TSP8:** `pad[0-5]`															

```
DiagPduEnv0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  PDU TEM Power Management Register tempwr (bf)
|  |  PDU TEM Power State temf (0) ... tem0 (15)
|  PDU EPU Crate Power States epupwr (bf)
|  |  Spare 10 bits spare0 (0-9)
|  |  PDU EPU Converter State epu2cvt (10) ... epu0cvt (12)
|  |  PDU EPU Crate Power State epu2pwr (13) ... epu0pwr (15)
|  PDU ADC Power Management Register acdpwr (bf)
|  |  Spare 13 bit field spare (0-12)
|  |  PDU ACD Power Converter Source cnvt (13)
```

```
|   |  PDU ACD Power Supply Source sup (14)
|   |  PDU ACD Power State acd (15)
|  TEM 3.3 Voltage ADC value temvolt[0-15] (bf)
|   |  ADC status bits stat (0-3)
|   |  ADC raw value val (4-15)
|  TEM Power Supply Temperature ADC value temps[0-15] (same bf)
|  EPU Voltage ADC value epuvolt[0-2] (same bf)
|  EPU Temperature ADC value eputemp[0-2] (same bf)
|  AEM DAQ Environmental dab (str)
|   |  AEM FREE Board 3.3 Sum sumv (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  DAQ Board Temperature dabt (same bf)
|   |  AEM FREE Board 28V Sum sumi (same bf)
|   |  DAQ Board 3.3 Voltage dabv (same bf)
|  Spare byte field pad[0-5] (fld)

typedef struct _LHK_DiagPduEnv0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* PDU TEM Power Management Register */
    LHK_PDUTEMPWRREG     tempwr;

    /* PDU EPU Crate Power States */
    LHK_PDUEPUCRATEPWR   epupwr;

    /* PDU ADC Power Management Register */
    LHK_ACDPWRREG        acdpwr;

    /* TEM 3.3 Voltage ADC value */
    LHK_TEM33V           temvolt[16];

    /* TEM Power Supply Temperature ADC value */
    LHK_TEMPSTEMP        temps[16];

    /* EPU Voltage ADC value */
    LHK_EPUV             epuvolt[3];

    /* EPU Temperature ADC value */
    LHK_EPUTEMP          eputemp[3];

    /* AEM DAQ Environmental */
    LHK_AEMDAQENV        dab;

    /* Spare byte field */
    unsigned char        pad[6];

} LHK_DiagPduEnv0_Tlm;
```

Fields:

`ACDPWRREG` **(10.6.2)**

Attribute(s):

Brief: "PDU ADC Power Management Register"

Instance(s): `acdpwr`

`AEMDAQENV` **(10.5.0)**

Attribute(s):

Brief: "AEM DAQ Environmental"

Instance(s): `dab`

`EPUTEMP` **(10.6.19)**

Attribute(s):

Brief: "EPU Temperature ADC value"

Instance(s): `eputemp`

`EPUV` **(10.6.20)**

Attribute(s):

Brief: "EPU Voltage ADC value"

Instance(s): `epuvolt`

`PDUEPUCRATEPWR` **(10.6.23)**

Attribute(s):

Brief: "PDU EPU Crate Power States"

Instance(s): `epupwr`

`PDUTEMPWRREG` **(10.6.24)**

Attribute(s):

Brief: "PDU TEM Power Management Register"

Instance(s): `tempwr`

`TEM33V` **(10.6.28)**

Attribute(s):

Brief: "TEM 3.3 Voltage ADC value"

Instance(s): `temvolt`

Describes a TEM 3.3V ADC value and status.

`TEMPSTEMP` **(10.6.31)**

Attribute(s):

Brief: "TEM Power Supply Temperature ADC value"

Instance(s): `temps`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.13 `DiagPduEnv1` (639/0x27F)

Context:

Description:

"Diagnostic PDU Environmental Packet 1" Telemetry Packet

PDU Packet 1

Layout:

Diagnostic PDU Environmental Packet 1 `(DiagPduEnv1)` **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x27F**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															

Diagnostic PDU Environmental Packet 1 (DiagPduEnv1) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** spares[0-5]															
014	**TEMPCBTEMP:** pcbt[0-15]															
034	**CALBSPTEMP:** bspt[0-15]															
054	**TSP8:** pad[0-31]															

```
DiagPduEnv1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM PCB Temperature pcbt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  CAL Baseplate Temperature bspt[0-15] (same bf)
|  Spare byte field pad[0-31] (fld)
```

```
typedef struct _LHK_DiagPduEnv1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM PCB Temperature */
    LHK_TEMPCBTEMP       pcbt[16];

    /* CAL Baseplate Temperature */
    LHK_CALBSPTEMP       bspt[16];

    /* Spare byte field */
    unsigned char        pad[32];

} LHK_DiagPduEnv1_Tlm;
```

Fields:

CALBSPTEMP **(10.6.16)**

Attribute(s):

Brief: "CAL Baseplate Temperature"

Instance(s): bspt

TEMPCBTEMP **(10.6.30)**

Attribute(s):

Brief: "TEM PCB Temperature"

Instance(s): `pcbt`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.14 `DiagPduEnv2` (640/0x280)

Context:

Description:

"Diagnostic PDU Environmental Packet 2" Telemetry Packet

PDU Packet 2

Layout:

Diagnostic PDU Environmental Packet 2 (DiagPduEnv2) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x280**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**VCHPDSHPTEMP:** vdi[0-11]															
02C	**VCHPXLHPTEMP:** vxi[0-11]															
044	**VCHPRSVRHTRTEMP:** vrh[0-11]															
05C	**TSP8:** pad[0-23]															

```
DiagPduEnv2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  VCHP-DSHP Interface Temperature vdi[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  VCHP-XLHP Interface Temperature vxi[0-11] (same bf)
|  VCHP Reservoir Heater Temperature vrh[0-11] (same bf)
|  Spare byte field pad[0-23] (fld)

typedef struct _LHK_DiagPduEnv2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
```

```
    unsigned char        spares[6];

    /* VCHP-DSHP Interface Temperature */
    LHK_VCHPDSHPTEMP     vdi[12];

    /* VCHP-XLHP Interface Temperature */
    LHK_VCHPXLHPTEMP     vxi[12];

    /* VCHP Reservoir Heater Temperature */
    LHK_VCHPRSVRHTRTEMP  vrh[12];

    /* Spare byte field */
    unsigned char        pad[24];

} LHK_DiagPduEnv2_Tlm;
```

Fields:

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

`VCHPDSHPTEMP` **(10.6.39)**

Attribute(s):

Brief: "VCHP-DSHP Interface Temperature"

Instance(s): `vdi`

`VCHPRSVRHTRTEMP` **(10.6.40)**

Attribute(s):

Brief: "VCHP Reservoir Heater Temperature"

Instance(s): `vrh`

`VCHPXLHPTEMP` **(10.6.41)**

Attribute(s):

Brief: "VCHP-XLHP Interface Temperature"

Instance(s): `vxi`

10.4.15 `DiagPduEnv3` (641/0x281)

Context:

Description:

"Diagnostic PDU Environmental Packet 3" Telemetry Packet

PDU Packet 3

Layout:

Diagnostic PDU Environmental Packet 3 (DiagPduEnv3) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x281										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**GRIDTEMP:** `gridt[0-11]`															
02C	**ACDSHELLTEMP:** `acdst[0-1]`															
030	**ACDPMTRAILTEMP:** `acdpr[0-3]`															
038	**ACDBEAGRIDTEMP:** `acdbg[0-1]`															
03C	**RADANHTRTEMP:** `rah[0-1]`															
040	**GRIDRADIFTEMP:** `grift[0-3]`															
048	**RADTEMP:** `rtemp[0-11]`															
060	**TSP8:** `pad[0-19]`															

```
DiagPduEnv3 (pkt)
|  Spare byte field spares[0-5] (fld)
|  Grid Temperature gridt[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  ADC Shell Temperature acdst[0-1] (same bf)
|  ADC PMT Rail Temperature acdpr[0-3] (same bf)
|  ACD BEA Grid Interface Temperature acdbg[0-1] (same bf)
|  Radiator Anitfreeze Heater Temperature rah[0-1] (same bf)
|  Grid Radiator Interface Temperature grift[0-3] (same bf)
|  Radiator Temperature rtemp[0-11] (same bf)
|  Spare byte field pad[0-19] (fld)

typedef struct _LHK_DiagPduEnv3_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* Grid Temperature */
```

```
    LHK_GRIDTEMP          gridt[12];

    /* ADC Shell Temperature */
    LHK_ACDSHELLTEMP      acdst[2];

    /* ADC PMT Rail Temperature */
    LHK_ACDPMTRAILTEMP    acdpr[4];

    /* ACD BEA Grid Interface Temperature */
    LHK_ACDBEAGRIDTEMP    acdbg[2];

    /* Radiator Anitfreeze Heater Temperature */
    LHK_RADANHTRTEMP      rah[2];

    /* Grid Radiator Interface Temperature */
    LHK_GRIDRADIFTEMP     grift[4];

    /* Radiator Temperature */
    LHK_RADTEMP           rtemp[12];

    /* Spare byte field */
    unsigned char         pad[20];

  } LHK_DiagPduEnv3_Tlm;
```

Fields:

`ACDBEAGRIDTEMP` **(10.6.0)**

Attribute(s):

Brief: "ACD BEA Grid Interface Temperature"

Instance(s): `acdbg`

`ACDPMTRAILTEMP` **(10.6.1)**

Attribute(s):

Brief: "ADC PMT Rail Temperature"

Instance(s): `acdpr`

`ACDSHELLTEMP` **(10.6.3)**

Attribute(s):

Brief: "ADC Shell Temperature"

Instance(s): `acdst`

`GRIDRADIFTEMP` **(10.6.21)**

Attribute(s):

Brief: "Grid Radiator Interface Temperature"

Instance(s): `grift`

`GRIDTEMP` **(10.6.22)**

Attribute(s):

Brief: "Grid Temperature"

Instance(s): `gridt`

`RADANHTRTEMP` **(10.6.25)**

Attribute(s):

Brief: "Radiator Anitfreeze Heater Temperature"

Instance(s): `rah`

`RADTEMP` **(10.6.26)**

Attribute(s):

Brief: "Radiator Temperature"

Instance(s): `rtemp`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.16 `DiagPduEnv4` (642/0x282)

Context:

Description:

"Diagnostic PDU Environmental Packet 4" Telemetry Packet

PDU Packet 4

Layout:

Diagnostic PDU Environmental Packet 4 (DiagPduEnv4) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x282										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**PDUTEMPWRREG:** `tempwr`															
016	**PDUEPUCRATEPWR:** `epupwr`															
018	**ACDPWRREG:** `acdpwr`															
01A	**TEM33V:** `temvolt[0-15]`															
03A	**TEMPSTEMP:** `temps[0-15]`															
05A	**EPUV:** `epuvolt[0-2]`															
060	**EPUTEMP:** `eputemp[0-2]`															
066	**TSP8:** `pad[0-13]`															

```
DiagPduEnv4 (pkt)
|  Spare byte field spares[0-5] (fld)
|  PDU TEM Power Management Register tempwr (bf)
|  |  PDU TEM Power State temf (0) ... tem0 (15)
|  PDU EPU Crate Power States epupwr (bf)
|  |  Spare 10 bits spare0 (0-9)
|  |  PDU EPU Converter State epu2cvt (10) ... epu0cvt (12)
|  |  PDU EPU Crate Power State epu2pwr (13) ... epu0pwr (15)
|  PDU ADC Power Management Register acdpwr (bf)
|  |  Spare 13 bit field spare (0-12)
|  |  PDU ACD Power Converter Source cnvt (13)
|  |  PDU ACD Power Supply Source sup (14)
|  |  PDU ACD Power State acd (15)
|  TEM 3.3 Voltage ADC value temvolt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  TEM Power Supply Temperature ADC value temps[0-15] (same bf)
|  EPU Voltage ADC value epuvolt[0-2] (same bf)
|  EPU Temperature ADC value eputemp[0-2] (same bf)
|  Spare byte field pad[0-13] (fld)
```

```
typedef struct _LHK_DiagPduEnv4_Tlm {
```

```
    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* PDU TEM Power Management Register */
    LHK_PDUTEMPWRREG     tempwr;

    /* PDU EPU Crate Power States */
    LHK_PDUEPUCRATEPWR   epupwr;

    /* PDU ADC Power Management Register */
    LHK_ACDPWRREG        acdpwr;

    /* TEM 3.3 Voltage ADC value */
    LHK_TEM33V           temvolt[16];

    /* TEM Power Supply Temperature ADC value */
    LHK_TEMPSTEMP        temps[16];

    /* EPU Voltage ADC value */
    LHK_EPUV             epuvolt[3];

    /* EPU Temperature ADC value */
    LHK_EPUTEMP          eputemp[3];

    /* Spare byte field */
    unsigned char        pad[14];

} LHK_DiagPduEnv4_Tlm;
```

Fields:

`ACDPWRREG` **(10.6.2)**

Attribute(s):

Brief: "PDU ADC Power Management Register"

Instance(s): `acdpwr`

`EPUTEMP` **(10.6.19)**

Attribute(s):

Brief: "EPU Temperature ADC value"

Instance(s): `eputemp`

`EPUV` **(10.6.20)**

Attribute(s):

Brief: "EPU Voltage ADC value"

Instance(s): `epuvolt`

`PDUEPUCRATEPWR` **(10.6.23)**

Attribute(s):

Brief: "PDU EPU Crate Power States"

Instance(s): `epupwr`

`PDUTEMPWRREG` **(10.6.24)**

Attribute(s):

Brief: "PDU TEM Power Management Register"

Instance(s): `tempwr`

`TEM33V` **(10.6.28)**

Attribute(s):

Brief: "TEM 3.3 Voltage ADC value"

Instance(s): `temvolt`

Describes a TEM 3.3V ADC value and status.

`TEMPSTEMP` **(10.6.31)**

Attribute(s):

Brief: "TEM Power Supply Temperature ADC value"

Instance(s): `temps`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.17 `DiagPduEnv5` (643/0x283)

Context:

Description:

"Diagnostic PDU Environmental Packet 5" Telemetry Packet

PDU Packet 5

Layout:

Diagnostic PDU Environmental Packet 5 (DiagPduEnv5) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x283**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TEMPCBTEMP:** pcbt[0-15]															
034	**CALBSPTEMP:** bspt[0-15]															
054	**TSP8:** pad[0-31]															

```
DiagPduEnv5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM PCB Temperature pcbt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  CAL Baseplate Temperature bspt[0-15] (same bf)
|  Spare byte field pad[0-31] (fld)
```

```
typedef struct _LHK_DiagPduEnv5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM PCB Temperature */
    LHK_TEMPCBTEMP       pcbt[16];

    /* CAL Baseplate Temperature */
    LHK_CALBSPTEMP       bspt[16];

    /* Spare byte field */
    unsigned char        pad[32];

} LHK_DiagPduEnv5_Tlm;
```

Fields:

`CALBSPTEMP` **(10.6.16)**

Attribute(s):

Brief: "CAL Baseplate Temperature"

Instance(s): `bspt`

`TEMPCBTEMP` **(10.6.30)**

Attribute(s):

Brief: "TEM PCB Temperature"

Instance(s): `pcbt`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.18 `DiagPduEnv6` (644/0x284)

Context:

Description:

"Diagnostic PDU Environmental Packet 6" Telemetry Packet

PDU Packet 6

Layout:

<table>
<tr><th colspan="17">Diagnostic PDU Environmental Packet 6 (DiagPduEnv6)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x284</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=109</td></tr>
<tr><td>006</td><td colspan="16">Timestamp</td></tr>
<tr><td>00E</td><td colspan="16">TSP8: spares[0-5]</td></tr>
<tr><td>014</td><td colspan="16">VCHPDSHPTEMP: vdi[0-11]</td></tr>
</table>

Diagnostic PDU Environmental Packet 6 (DiagPduEnv6) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
02C	**VCHPXLHPTEMP:** vxi[0-11]															
044	**VCHPRSVRHTRTEMP:** vrh[0-11]															
05C	**TSP8:** pad[0-23]															

```
DiagPduEnv6 (pkt)
|  Spare byte field spares[0-5] (fld)
|  VCHP-DSHP Interface Temperature vdi[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  VCHP-XLHP Interface Temperature vxi[0-11] (same bf)
|  VCHP Reservoir Heater Temperature vrh[0-11] (same bf)
|  Spare byte field pad[0-23] (fld)
```

```
typedef struct _LHK_DiagPduEnv6_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* VCHP-DSHP Interface Temperature */
    LHK_VCHPDSHPTEMP     vdi[12];

    /* VCHP-XLHP Interface Temperature */
    LHK_VCHPXLHPTEMP     vxi[12];

    /* VCHP Reservoir Heater Temperature */
    LHK_VCHPRSVRHTRTEMP  vrh[12];

    /* Spare byte field */
    unsigned char        pad[24];

} LHK_DiagPduEnv6_Tlm;
```

Fields:

TSP8 **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): pad, spares

VCHPDSHPTEMP **(10.6.39)**

Attribute(s):

Brief: "VCHP-DSHP Interface Temperature"

Instance(s): `vdi`

`VCHPRSVRHTRTEMP` **(10.6.40)**

Attribute(s):

Brief: "VCHP Reservoir Heater Temperature"

Instance(s): `vrh`

`VCHPXLHPTEMP` **(10.6.41)**

Attribute(s):

Brief: "VCHP-XLHP Interface Temperature"

Instance(s): `vxi`

10.4.19 `DiagPduEnv7` (645/0x285)

Context:

Description:

"Diagnostic PDU Environmental Packet 7" Telemetry Packet

PDU Packet 7

Layout:

Diagnostic PDU Environmental Packet 7 (`DiagPduEnv7`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x285**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**GRIDTEMP:** `gridt[0-11]`															
02C	**ACDSHELLTEMP:** `acdst[0-1]`															
030	**ACDPMTRAILTEMP:** `acdpr[0-3]`															
038	**ACDBEAGRIDTEMP:** `acdbg[0-1]`															
03C	**RADANHTRTEMP:** `rah[0-1]`															
040	**GRIDRADIFTEMP:** `grift[0-3]`															
048	**RADTEMP:** `rtemp[0-11]`															

Diagnostic PDU Environmental Packet 7 (DiagPduEnv7) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
060	**TSP8:** pad[0-19]															

```
DiagPduEnv7 (pkt)
|  Spare byte field spares[0-5] (fld)
|  Grid Temperature gridt[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  ADC Shell Temperature acdst[0-1] (same bf)
|  ADC PMT Rail Temperature acdpr[0-3] (same bf)
|  ACD BEA Grid Interface Temperature acdbg[0-1] (same bf)
|  Radiator Anitfreeze Heater Temperature rah[0-1] (same bf)
|  Grid Radiator Interface Temperature grift[0-3] (same bf)
|  Radiator Temperature rtemp[0-11] (same bf)
|  Spare byte field pad[0-19] (fld)
```

```
typedef struct _LHK_DiagPduEnv7_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* Grid Temperature */
    LHK_GRIDTEMP         gridt[12];

    /* ADC Shell Temperature */
    LHK_ACDSHELLTEMP     acdst[2];

    /* ADC PMT Rail Temperature */
    LHK_ACDPMTRAILTEMP   acdpr[4];

    /* ACD BEA Grid Interface Temperature */
    LHK_ACDBEAGRIDTEMP   acdbg[2];

    /* Radiator Anitfreeze Heater Temperature */
    LHK_RADANHTRTEMP     rah[2];

    /* Grid Radiator Interface Temperature */
    LHK_GRIDRADIFTEMP    grift[4];

    /* Radiator Temperature */
    LHK_RADTEMP          rtemp[12];

    /* Spare byte field */
    unsigned char        pad[20];

} LHK_DiagPduEnv7_Tlm;
```

Fields:

`ACDBEAGRIDTEMP` **(10.6.0)**

Attribute(s):

Brief: "ACD BEA Grid Interface Temperature"

Instance(s): `acdbg`

`ACDPMTRAILTEMP` **(10.6.1)**

Attribute(s):

Brief: "ADC PMT Rail Temperature"

Instance(s): `acdpr`

`ACDSHELLTEMP` **(10.6.3)**

Attribute(s):

Brief: "ADC Shell Temperature"

Instance(s): `acdst`

`GRIDRADIFTEMP` **(10.6.21)**

Attribute(s):

Brief: "Grid Radiator Interface Temperature"

Instance(s): `grift`

`GRIDTEMP` **(10.6.22)**

Attribute(s):

Brief: "Grid Temperature"

Instance(s): `gridt`

`RADANHTRTEMP` **(10.6.25)**

Attribute(s):

Brief: "Radiator Anitfreeze Heater Temperature"

Instance(s): `rah`

`RADTEMP` **(10.6.26)**

Attribute(s):

Brief: "Radiator Temperature"

Instance(s): `rtemp`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.20 `DiagTemEnvPwr0` (624/0x270)

Context:

Description:

"Diagnostic TEM Power Packet 0" Telemetry Packet

Contains power specific ADC values for TEMs 0, 1, and 2.

Layout:

Diagnostic TEM Power Packet 0 (`DiagTemEnvPwr0`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x270**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `t0m0`															
024	**TMUX1:** `t0m1`															
034	**TMUX0:** `t1m0`															
044	**TMUX1:** `t1m1`															
054	**TMUX0:** `t2m0`															
064	**TMUX1:** `t2m1`															

```
DiagTemEnvPwr0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t0m0 (str)
```

```
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0–3)
│   │   │   ADC raw value val (4–15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs t0m1 (str)
│   │   CAL 3.3 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0–3)
│   │   │   ADC raw value val (4–15)
│   │   CAL 3.3 Current ADC value adc1 (same bf)
│   │   CAL 3.3 Voltage ADC value adc2 (same bf)
│   │   CAL 3.3 Current ADC value adc3 (same bf)
│   │   CAL Bias Voltage ADC value adc4 (same bf)
│   │   CAL Bias Current ADC value adc5 (same bf)
│   │   TEM 3.3 Voltage ADC value adc6 (same bf)
│   │   TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs t1m0 (str)
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0–3)
│   │   │   ADC raw value val (4–15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs t1m1 (str)
│   │   CAL 3.3 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0–3)
│   │   │   ADC raw value val (4–15)
│   │   CAL 3.3 Current ADC value adc1 (same bf)
│   │   CAL 3.3 Voltage ADC value adc2 (same bf)
│   │   CAL 3.3 Current ADC value adc3 (same bf)
│   │   CAL Bias Voltage ADC value adc4 (same bf)
│   │   CAL Bias Current ADC value adc5 (same bf)
│   │   TEM 3.3 Voltage ADC value adc6 (same bf)
│   │   TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs t2m0 (str)
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0–3)
│   │   │   ADC raw value val (4–15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
```

```
|  TEM mux channel 1 environmental ADCs t2m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_DiagTemEnvPwr0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t0m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t0m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t1m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t1m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t2m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t2m1;

} LHK_DiagTemEnvPwr0_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t0m0`, `t1m0`, `t2m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t0m1`, `t1m1`, `t2m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.21 `DiagTemEnvPwr1` (625/0x271)

Context:

Description:

"Diagnostic TEM Power Packet 1" Telemetry Packet

Contains power specific ADC values for TEMs 3,4, and 5.

Layout:

Diagnostic TEM Power Packet 1 (`DiagTemEnvPwr1`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x271**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `t3m0`															
024	**TMUX1:** `t3m1`															
034	**TMUX0:** `t4m0`															
044	**TMUX1:** `t4m1`															
054	**TMUX0:** `t5m0`															
064	**TMUX1:** `t5m1`															

```
DiagTemEnvPwr1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t3m0 (str)
```

```
│   │  TKR 2.5 Voltage ADC value adc0 (bf)
│   │  │  ADC status bits stat (0−3)
│   │  │  ADC raw value val (4−15)
│   │  TKR 2.5 Current ADC value adc1 (same bf)
│   │  TKR 1.5 Voltage ADC value adc2 (same bf)
│   │  TKR 1.5 Current ADC value adc3 (same bf)
│   │  TKR 2.5 Voltage ADC value adc4 (same bf)
│   │  TKR 2.5 Current ADC value adc5 (same bf)
│   │  TKR Bias Voltage ADC value adc6 (same bf)
│   │  TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs t3m1 (str)
│   │  CAL 3.3 Voltage ADC value adc0 (bf)
│   │  │  ADC status bits stat (0−3)
│   │  │  ADC raw value val (4−15)
│   │  CAL 3.3 Current ADC value adc1 (same bf)
│   │  CAL 3.3 Voltage ADC value adc2 (same bf)
│   │  CAL 3.3 Current ADC value adc3 (same bf)
│   │  CAL Bias Voltage ADC value adc4 (same bf)
│   │  CAL Bias Current ADC value adc5 (same bf)
│   │  TEM 3.3 Voltage ADC value adc6 (same bf)
│   │  TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs t4m0 (str)
│   │  TKR 2.5 Voltage ADC value adc0 (bf)
│   │  │  ADC status bits stat (0−3)
│   │  │  ADC raw value val (4−15)
│   │  TKR 2.5 Current ADC value adc1 (same bf)
│   │  TKR 1.5 Voltage ADC value adc2 (same bf)
│   │  TKR 1.5 Current ADC value adc3 (same bf)
│   │  TKR 2.5 Voltage ADC value adc4 (same bf)
│   │  TKR 2.5 Current ADC value adc5 (same bf)
│   │  TKR Bias Voltage ADC value adc6 (same bf)
│   │  TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs t4m1 (str)
│   │  CAL 3.3 Voltage ADC value adc0 (bf)
│   │  │  ADC status bits stat (0−3)
│   │  │  ADC raw value val (4−15)
│   │  CAL 3.3 Current ADC value adc1 (same bf)
│   │  CAL 3.3 Voltage ADC value adc2 (same bf)
│   │  CAL 3.3 Current ADC value adc3 (same bf)
│   │  CAL Bias Voltage ADC value adc4 (same bf)
│   │  CAL Bias Current ADC value adc5 (same bf)
│   │  TEM 3.3 Voltage ADC value adc6 (same bf)
│   │  TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs t5m0 (str)
│   │  TKR 2.5 Voltage ADC value adc0 (bf)
│   │  │  ADC status bits stat (0−3)
│   │  │  ADC raw value val (4−15)
│   │  TKR 2.5 Current ADC value adc1 (same bf)
│   │  TKR 1.5 Voltage ADC value adc2 (same bf)
│   │  TKR 1.5 Current ADC value adc3 (same bf)
│   │  TKR 2.5 Voltage ADC value adc4 (same bf)
│   │  TKR 2.5 Current ADC value adc5 (same bf)
│   │  TKR Bias Voltage ADC value adc6 (same bf)
│   │  TKR Bias Current ADC value adc7 (same bf)
```

```
|  TEM mux channel 1 environmental ADCs t5m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_DiagTemEnvPwr1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t3m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t3m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t4m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t4m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t5m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t5m1;

} LHK_DiagTemEnvPwr1_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t3m0`, `t4m0`, `t5m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t3m1`, `t4m1`, `t5m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.22 `DiagTemEnvPwr2` (626/0x272)

Context:

Description:

"Diagnostic TEM Power Packet 2" Telemetry Packet

Contains power specific ADC values for TEMs 6, 7, and 8.

Layout:

Diagnostic TEM Power Packet 2 (`DiagTemEnvPwr2`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x272**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `t6m0`															
024	**TMUX1:** `t6m1`															
034	**TMUX0:** `t7m0`															
044	**TMUX1:** `t7m1`															
054	**TMUX0:** `t8m0`															
064	**TMUX1:** `t8m1`															

```
DiagTemEnvPwr2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t6m0 (str)
```

```
│  │  TKR 2.5 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0−3)
│  │  │  ADC raw value val (4−15)
│  │  TKR 2.5 Current ADC value adc1 (same bf)
│  │  TKR 1.5 Voltage ADC value adc2 (same bf)
│  │  TKR 1.5 Current ADC value adc3 (same bf)
│  │  TKR 2.5 Voltage ADC value adc4 (same bf)
│  │  TKR 2.5 Current ADC value adc5 (same bf)
│  │  TKR Bias Voltage ADC value adc6 (same bf)
│  │  TKR Bias Current ADC value adc7 (same bf)
│  TEM mux channel 1 environmental ADCs t6m1 (str)
│  │  CAL 3.3 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0−3)
│  │  │  ADC raw value val (4−15)
│  │  CAL 3.3 Current ADC value adc1 (same bf)
│  │  CAL 3.3 Voltage ADC value adc2 (same bf)
│  │  CAL 3.3 Current ADC value adc3 (same bf)
│  │  CAL Bias Voltage ADC value adc4 (same bf)
│  │  CAL Bias Current ADC value adc5 (same bf)
│  │  TEM 3.3 Voltage ADC value adc6 (same bf)
│  │  TEM 3.3 Current ADC value adc7 (same bf)
│  TEM mux channel 0 environmental ADCs t7m0 (str)
│  │  TKR 2.5 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0−3)
│  │  │  ADC raw value val (4−15)
│  │  TKR 2.5 Current ADC value adc1 (same bf)
│  │  TKR 1.5 Voltage ADC value adc2 (same bf)
│  │  TKR 1.5 Current ADC value adc3 (same bf)
│  │  TKR 2.5 Voltage ADC value adc4 (same bf)
│  │  TKR 2.5 Current ADC value adc5 (same bf)
│  │  TKR Bias Voltage ADC value adc6 (same bf)
│  │  TKR Bias Current ADC value adc7 (same bf)
│  TEM mux channel 1 environmental ADCs t7m1 (str)
│  │  CAL 3.3 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0−3)
│  │  │  ADC raw value val (4−15)
│  │  CAL 3.3 Current ADC value adc1 (same bf)
│  │  CAL 3.3 Voltage ADC value adc2 (same bf)
│  │  CAL 3.3 Current ADC value adc3 (same bf)
│  │  CAL Bias Voltage ADC value adc4 (same bf)
│  │  CAL Bias Current ADC value adc5 (same bf)
│  │  TEM 3.3 Voltage ADC value adc6 (same bf)
│  │  TEM 3.3 Current ADC value adc7 (same bf)
│  TEM mux channel 0 environmental ADCs t8m0 (str)
│  │  TKR 2.5 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0−3)
│  │  │  ADC raw value val (4−15)
│  │  TKR 2.5 Current ADC value adc1 (same bf)
│  │  TKR 1.5 Voltage ADC value adc2 (same bf)
│  │  TKR 1.5 Current ADC value adc3 (same bf)
│  │  TKR 2.5 Voltage ADC value adc4 (same bf)
│  │  TKR 2.5 Current ADC value adc5 (same bf)
│  │  TKR Bias Voltage ADC value adc6 (same bf)
│  │  TKR Bias Current ADC value adc7 (same bf)
```

```
|  TEM mux channel 1 environmental ADCs t8m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_DiagTemEnvPwr2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t6m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t6m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t7m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t7m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t8m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t8m1;

} LHK_DiagTemEnvPwr2_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t6m0`, `t7m0`, `t8m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t6m1`, `t7m1`, `t8m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.23 `DiagTemEnvPwr3` (627/0x273)

Context:

Description:

"Diagnostic TEM Power Packet 3" Telemetry Packet

Contains power specific ADC values for TEMs 9, 10, and 11.

Layout:

Diagnostic TEM Power Packet 3 (`DiagTemEnvPwr3`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x273**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `t9m0`															
024	**TMUX1:** `t9m1`															
034	**TMUX0:** `tAm0`															
044	**TMUX1:** `tAm1`															
054	**TMUX0:** `tBm0`															
064	**TMUX1:** `tBm1`															

```
DiagTemEnvPwr3 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t9m0 (str)
```

```
|   |   TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |   ADC status bits stat (0−3)
|   |   |   ADC raw value val (4−15)
|   |   TKR 2.5 Current ADC value adc1 (same bf)
|   |   TKR 1.5 Voltage ADC value adc2 (same bf)
|   |   TKR 1.5 Current ADC value adc3 (same bf)
|   |   TKR 2.5 Voltage ADC value adc4 (same bf)
|   |   TKR 2.5 Current ADC value adc5 (same bf)
|   |   TKR Bias Voltage ADC value adc6 (same bf)
|   |   TKR Bias Current ADC value adc7 (same bf)
|   TEM mux channel 1 environmental ADCs t9m1 (str)
|   |   CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |   ADC status bits stat (0−3)
|   |   |   ADC raw value val (4−15)
|   |   CAL 3.3 Current ADC value adc1 (same bf)
|   |   CAL 3.3 Voltage ADC value adc2 (same bf)
|   |   CAL 3.3 Current ADC value adc3 (same bf)
|   |   CAL Bias Voltage ADC value adc4 (same bf)
|   |   CAL Bias Current ADC value adc5 (same bf)
|   |   TEM 3.3 Voltage ADC value adc6 (same bf)
|   |   TEM 3.3 Current ADC value adc7 (same bf)
|   TEM mux channel 0 environmental ADCs tAm0 (str)
|   |   TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |   ADC status bits stat (0−3)
|   |   |   ADC raw value val (4−15)
|   |   TKR 2.5 Current ADC value adc1 (same bf)
|   |   TKR 1.5 Voltage ADC value adc2 (same bf)
|   |   TKR 1.5 Current ADC value adc3 (same bf)
|   |   TKR 2.5 Voltage ADC value adc4 (same bf)
|   |   TKR 2.5 Current ADC value adc5 (same bf)
|   |   TKR Bias Voltage ADC value adc6 (same bf)
|   |   TKR Bias Current ADC value adc7 (same bf)
|   TEM mux channel 1 environmental ADCs tAm1 (str)
|   |   CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |   ADC status bits stat (0−3)
|   |   |   ADC raw value val (4−15)
|   |   CAL 3.3 Current ADC value adc1 (same bf)
|   |   CAL 3.3 Voltage ADC value adc2 (same bf)
|   |   CAL 3.3 Current ADC value adc3 (same bf)
|   |   CAL Bias Voltage ADC value adc4 (same bf)
|   |   CAL Bias Current ADC value adc5 (same bf)
|   |   TEM 3.3 Voltage ADC value adc6 (same bf)
|   |   TEM 3.3 Current ADC value adc7 (same bf)
|   TEM mux channel 0 environmental ADCs tBm0 (str)
|   |   TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |   ADC status bits stat (0−3)
|   |   |   ADC raw value val (4−15)
|   |   TKR 2.5 Current ADC value adc1 (same bf)
|   |   TKR 1.5 Voltage ADC value adc2 (same bf)
|   |   TKR 1.5 Current ADC value adc3 (same bf)
|   |   TKR 2.5 Voltage ADC value adc4 (same bf)
|   |   TKR 2.5 Current ADC value adc5 (same bf)
|   |   TKR Bias Voltage ADC value adc6 (same bf)
|   |   TKR Bias Current ADC value adc7 (same bf)
```

```
|  TEM mux channel 1 environmental ADCs tBm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_DiagTemEnvPwr3_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t9m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t9m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tAm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tAm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tBm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tBm1;

} LHK_DiagTemEnvPwr3_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t9m0`, `tAm0`, `tBm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t9m1`, `tAm1`, `tBm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.24 `DiagTemEnvPwr4` (628/0x274)

Context:

Description:

"Diagnostic TEM Power Packet 4" Telemetry Packet

Contains power specific ADC values for TEMs 12, 13, and 14.

Layout:

Diagnostic TEM Power Packet 4 (DiagTemEnvPwr4) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x274**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `tCm0`															
024	**TMUX1:** `tCm1`															
034	**TMUX0:** `tDm0`															
044	**TMUX1:** `tDm1`															
054	**TMUX0:** `tEm0`															
064	**TMUX1:** `tEm1`															

```
DiagTemEnvPwr4 (pkt)
│  Spare byte field spares[0-5] (fld)
│  TEM mux channel 0 environmental ADCs tCm0 (str)
```

```
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0−3)
│   │   │   ADC raw value val (4−15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs tCm1 (str)
│   │   CAL 3.3 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0−3)
│   │   │   ADC raw value val (4−15)
│   │   CAL 3.3 Current ADC value adc1 (same bf)
│   │   CAL 3.3 Voltage ADC value adc2 (same bf)
│   │   CAL 3.3 Current ADC value adc3 (same bf)
│   │   CAL Bias Voltage ADC value adc4 (same bf)
│   │   CAL Bias Current ADC value adc5 (same bf)
│   │   TEM 3.3 Voltage ADC value adc6 (same bf)
│   │   TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs tDm0 (str)
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0−3)
│   │   │   ADC raw value val (4−15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
│   TEM mux channel 1 environmental ADCs tDm1 (str)
│   │   CAL 3.3 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0−3)
│   │   │   ADC raw value val (4−15)
│   │   CAL 3.3 Current ADC value adc1 (same bf)
│   │   CAL 3.3 Voltage ADC value adc2 (same bf)
│   │   CAL 3.3 Current ADC value adc3 (same bf)
│   │   CAL Bias Voltage ADC value adc4 (same bf)
│   │   CAL Bias Current ADC value adc5 (same bf)
│   │   TEM 3.3 Voltage ADC value adc6 (same bf)
│   │   TEM 3.3 Current ADC value adc7 (same bf)
│   TEM mux channel 0 environmental ADCs tEm0 (str)
│   │   TKR 2.5 Voltage ADC value adc0 (bf)
│   │   │   ADC status bits stat (0−3)
│   │   │   ADC raw value val (4−15)
│   │   TKR 2.5 Current ADC value adc1 (same bf)
│   │   TKR 1.5 Voltage ADC value adc2 (same bf)
│   │   TKR 1.5 Current ADC value adc3 (same bf)
│   │   TKR 2.5 Voltage ADC value adc4 (same bf)
│   │   TKR 2.5 Current ADC value adc5 (same bf)
│   │   TKR Bias Voltage ADC value adc6 (same bf)
│   │   TKR Bias Current ADC value adc7 (same bf)
```

```
|  TEM mux channel 1 environmental ADCs tEm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_DiagTemEnvPwr4_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tCm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tCm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tDm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tDm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tEm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tEm1;

} LHK_DiagTemEnvPwr4_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `tCm0`, `tDm0`, `tEm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `tCm1`, `tDm1`, `tEm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.25 `DiagTemEnvPwr5` (629/0x275)

Context:

Description:

"Diagnostic TEM Power Packet 5" Telemetry Packet

Contains powr specific ADC values for TEM 15.

Layout:

Diagnostic TEM Power Packet 5 (DiagTemEnvPwr5) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x275**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `tFm0`															
024	**TMUX1:** `tFm1`															
034	**TSP16:** `pad[0-31]`															

```
DiagTemEnvPwr5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs tFm0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
```

```
│  │  TKR 1.5 Voltage ADC value adc2 (same bf)
│  │  TKR 1.5 Current ADC value adc3 (same bf)
│  │  TKR 2.5 Voltage ADC value adc4 (same bf)
│  │  TKR 2.5 Current ADC value adc5 (same bf)
│  │  TKR Bias Voltage ADC value adc6 (same bf)
│  │  TKR Bias Current ADC value adc7 (same bf)
│  TEM mux channel 1 environmental ADCs tFm1 (str)
│  │  CAL 3.3 Voltage ADC value adc0 (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  │  CAL 3.3 Current ADC value adc1 (same bf)
│  │  CAL 3.3 Voltage ADC value adc2 (same bf)
│  │  CAL 3.3 Current ADC value adc3 (same bf)
│  │  CAL Bias Voltage ADC value adc4 (same bf)
│  │  CAL Bias Current ADC value adc5 (same bf)
│  │  TEM 3.3 Voltage ADC value adc6 (same bf)
│  │  TEM 3.3 Current ADC value adc7 (same bf)
│  Spare 16 bit field pad[0-31] (fld)
```

```
typedef struct _LHK_DiagTemEnvPwr5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tFm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tFm1;

    /* Spare 16 bit field */
    unsigned short       pad[32];

} LHK_DiagTemEnvPwr5_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `tFm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `tFm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `pad`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.26 `DiagTemEnvTemp0` (630/0x276)

Context:

Description:

"Diagnostic TEM Temperature Packet 0" Telemetry Packet

Contains temperature specific ADC values for TEMs 0 and 1.

Layout:

Diagnostic TEM Temperature Packet 0 (`DiagTemEnvTemp0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x276**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t0m2`															
024	**TMUX3:** `t0m3`															
034	**TMUX4:** `t0m4`															

Diagnostic TEM Temperature Packet 0 (DiagTemEnvTemp0) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
044	**TMUX2:** `t1m0`															
054	**TMUX3:** `t1m1`															
064	**TMUX4:** `t1m2`															

```
DiagTemEnvTemp0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t0m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t0m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t0m4 (same str)
|  TEM mux channel 2 environmental ADCs t1m0 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t1m1 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t1m2 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp0_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t0m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t0m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t0m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t1m0;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t1m1;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t1m2;
```

```
} LHK_DiagTemEnvTemp0_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t0m2`, `t1m0`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t0m3`, `t1m1`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t0m4`, `t1m2`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.27 `DiagTemEnvTemp1` (631/0x277)

Context:

Description:

"Diagnostic TEM Temperature Packet 1" Telemetry Packet

Contains temperature specific ADC values for TEMs 2 and 3.

Layout:

Diagnostic TEM Temperature Packet 1 (`DiagTemEnvTemp1`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x277**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t2m2`															
024	**TMUX3:** `t2m3`															
034	**TMUX4:** `t2m4`															
044	**TMUX2:** `t3m2`															
054	**TMUX3:** `t3m3`															
064	**TMUX4:** `t3m4`															

```
DiagTemEnvTemp1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t2m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t2m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t2m4 (same str)
|  TEM mux channel 2 environmental ADCs t3m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t3m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t3m4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp1_Tlm {

    /* Telemetry header */
```

```
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t2m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t2m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t2m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t3m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t3m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t3m4;

} LHK_DiagTemEnvTemp1_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t2m2`, `t3m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t2m3`, `t3m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t2m4`, `t3m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.28 `DiagTemEnvTemp2` (632/0x278)

Context:

Description:

"Diagnostic TEM Temperature Packet 2" Telemetry Packet

Contains temperature specific ADC values for TEMs 4 and 5.

Layout:

Diagnostic TEM Temperature Packet 2 (DiagTemEnvTemp2) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x278**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t4m2`															
024	**TMUX3:** `t4m3`															
034	**TMUX4:** `t4m4`															
044	**TMUX2:** `t5m2`															
054	**TMUX3:** `t5m3`															
064	**TMUX4:** `t5m4`															

```
DiagTemEnvTemp2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t4m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t4m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
```

```
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t4m4 (same str)
|  TEM mux channel 2 environmental ADCs t5m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t5m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t5m4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t4m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t4m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t4m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t5m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t5m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t5m4;

} LHK_DiagTemEnvTemp2_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t4m2`, `t5m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t4m3`, `t5m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t4m4`, `t5m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.29 `DiagTemEnvTemp3` (633/0x279)

Context:

Description:

"Diagnostic TEM Temperature Packet 3" Telemetry Packet

Contains temperature specific ADC values for TEMs 6 and 7.

Layout:

Diagnostic TEM Temperature Packet 3 (`DiagTemEnvTemp3`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x279**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															

Diagnostic TEM Temperature Packet 3 (DiagTemEnvTemp3) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t6m2`															
024	**TMUX3:** `t6m3`															
034	**TMUX4:** `t6m4`															
044	**TMUX2:** `t7m2`															
054	**TMUX3:** `t7m3`															
064	**TMUX4:** `t7m4`															

```
DiagTemEnvTemp3 (pkt)
│  Spare byte field spares[0-5] (fld)
│  TEM mux channel 2 environmental ADCs t6m2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs t6m3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs t6m4 (same str)
│  TEM mux channel 2 environmental ADCs t7m2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs t7m3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs t7m4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp3_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t6m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t6m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t6m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t7m2;

    /* TEM mux channel 3 environmental ADCs */
```

```
      LHK_TMUX3            t7m3;

      /* TEM mux channel 4 environmental ADCs */
      LHK_TMUX4            t7m4;

   } LHK_DiagTemEnvTemp3_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t6m2`, `t7m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t6m3`, `t7m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t6m4`, `t7m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.30 `DiagTemEnvTemp4` (634/0x27A)

Context:

Description:

"Diagnostic TEM Temperature Packet 4" Telemetry Packet

Contains temperature specific ADC values for TEMs 8 and 9.

Layout:

Diagnostic TEM Temperature Packet 4 (DiagTemEnvTemp4) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x27A**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** t8m2															
024	**TMUX3:** t8m3															
034	**TMUX4:** t8m4															
044	**TMUX2:** t9m2															
054	**TMUX3:** t9m3															
064	**TMUX4:** t9m4															

```
DiagTemEnvTemp4 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t8m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t8m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t8m4 (same str)
|  TEM mux channel 2 environmental ADCs t9m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t9m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t9m4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp4_Tlm {

    /* Telemetry header */
```

```
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t8m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t8m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t8m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t9m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t9m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t9m4;

} LHK_DiagTemEnvTemp4_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t8m2`, `t9m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t8m3`, `t9m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t8m4`, `t9m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.31 `DiagTemEnvTemp5` (635/0x27B)

Context:

Description:

"Diagnostic TEM Temperature Packet 5" Telemetry Packet

Contains temperature specific ADC values for TEMs A and B.

Layout:

Diagnostic TEM Temperature Packet 5 (DiagTemEnvTemp5) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x27B**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `tAm2`															
024	**TMUX3:** `tAm3`															
034	**TMUX4:** `tAm4`															
044	**TMUX2:** `tBm2`															
054	**TMUX3:** `tBm3`															
064	**TMUX4:** `tBm4`															

```
DiagTemEnvTemp5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs tAm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tAm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
```

```
|   |   |   ADC status bits stat (0-3)
|   |   |   ADC raw value val (4-15)
|   TEM mux channel 4 environmental ADCs tAm4 (same str)
|   TEM mux channel 2 environmental ADCs tBm2 (str)
|   |   CAL AFFE temperature ADC value adcs[0-7] (bf)
|   |   |   ADC status bits stat (0-3)
|   |   |   ADC raw value val (4-15)
|   TEM mux channel 3 environmental ADCs tBm3 (str)
|   |   TKR Cable Temperature ADC value adcs[0-7] (bf)
|   |   |   ADC status bits stat (0-3)
|   |   |   ADC raw value val (4-15)
|   TEM mux channel 4 environmental ADCs tBm4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tAm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tAm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tAm4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tBm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tBm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tBm4;

} LHK_DiagTemEnvTemp5_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tAm2`, `tBm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tAm3`, `tBm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tAm4`, `tBm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.32 `DiagTemEnvTemp6` (636/0x27C)

Context:

Description:

"Diagnostic TEM Temperature Packet 6" Telemetry Packet

Contains temperature specific ADC values for TEMs C and D.

Layout:

<table>
<tr><th colspan="17">Diagnostic TEM Temperature Packet 6 (DiagTemEnvTemp6)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x27C</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=109</td></tr>
<tr><td>006</td><td colspan="16">Timestamp</td></tr>
</table>

Diagnostic TEM Temperature Packet 6 (DiagTemEnvTemp6) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** tCm2															
024	**TMUX3:** tCm3															
034	**TMUX4:** tCm4															
044	**TMUX2:** tDm2															
054	**TMUX3:** tDm3															
064	**TMUX4:** tDm4															

```
DiagTemEnvTemp6 (pkt)
│  Spare byte field spares[0-5] (fld)
│  TEM mux channel 2 environmental ADCs tCm2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs tCm3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs tCm4 (same str)
│  TEM mux channel 2 environmental ADCs tDm2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs tDm3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs tDm4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp6_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tCm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tCm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tCm4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tDm2;

    /* TEM mux channel 3 environmental ADCs */
```

```
    LHK_TMUX3            tDm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tDm4;

} LHK_DiagTemEnvTemp6_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tCm2`, `tDm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tCm3`, `tDm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tCm4`, `tDm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.33 `DiagTemEnvTemp7` (637/0x27D)

Context:

Description:

"Diagnostic TEM Temperature Packet 7" Telemetry Packet

Contains temperature specific ADC values for TEMs E and F.

Layout:

Diagnostic TEM Temperature Packet 7 (DiagTemEnvTemp7) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x27D										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `tEm3`															
024	**TMUX3:** `tEm4`															
034	**TMUX4:** `tEm5`															
044	**TMUX2:** `tFm2`															
054	**TMUX3:** `tFm3`															
064	**TMUX4:** `tFm4`															

```
DiagTemEnvTemp7 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs tEm3 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tEm4 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tEm5 (same str)
|  TEM mux channel 2 environmental ADCs tFm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tFm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tFm4 (same str)
```

```
typedef struct _LHK_DiagTemEnvTemp7_Tlm {

    /* Telemetry header */
```

```
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           tEm3;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           tEm4;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           tEm5;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           tFm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           tFm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           tFm4;

} LHK_DiagTemEnvTemp7_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tEm3`, `tFm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tEm4`, `tFm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tEm5`, `tFm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

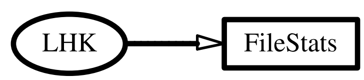
Attribute(s):

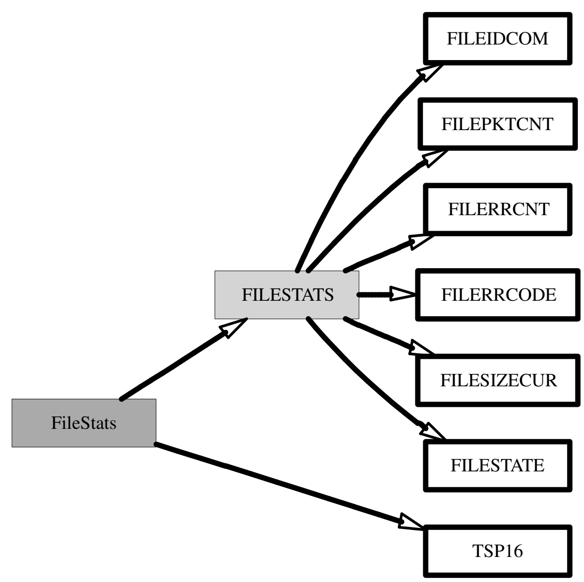
Brief: "Spare byte field"

Instance(s): `spares`

10.4.34 `FileStats` (554/0x22A)

Context:





Description:

"File System Statistics Packet" Telemetry Packet

Contains FILE statistics for each CPU.

Layout:

File System Statistics Packet (FileStats) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x22A**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spares[0-2]															
014	**FILESTATS:** siu															
02C	**FILESTATS:** epu0															
044	**FILESTATS:** epu1															
05C	**FILESTATS:** epu2															

```
FileStats (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  FILE Upload Statistics siu (str)
|  |  FILE Upload State state (fld)
|  |  File Size Current cursize (fld)
|  |  FILE Packet Count pktcnt (fld)
|  |  FILE Error Code errcode (fld)
|  |  FILE Error Count errcnt (fld)
|  |  FILE ID Commit comid (fld)
|  FILE Upload Statistics epu0 (same str)
|  FILE Upload Statistics epu1 (same str)
|  FILE Upload Statistics epu2 (same str)
```

```
typedef struct _LHK_FileStats_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* FILE Upload Statistics */
    LHK_FILESTATS        siu;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu0;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu1;

    /* FILE Upload Statistics */
    LHK_FILESTATS        epu2;

} LHK_FileStats_Tlm;
```

Fields:

FILESTATS **(10.5.4)**

Attribute(s):

Brief: "FILE Upload Statistics"

Instance(s): `epu0`, `epu1`, `epu2`, `siu`

Describes the FILE upload current state.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.35 `Lrs0` (551/0x227)

Context:

Description:

"Low-rate Science Packet" Telemetry Packet

Contains TEM dead-time counters and GEM livetime counter

Layout:

Low-rate Science Packet (Lrs0) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x227**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** `spares[0-2]`															
014	**TIMESEC:** `gemsec`															
018	**TIMESUBSEC:** `gemsub`															
01C	**GEMLRS:** `gemlrs`															
02C	**TIMESEC:** `temsec`															
030	**TIMESUBSEC:** `temsub`															
034	**TEMDEADTIMEREG:** `temlrs[0-15]`															

```
Lrs0 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  Timestamp seconds gemsec (fld)
|  Timestamp subseconds gemsub (fld)
|  GEM Low-Rate Science Counter gemlrs (str)
|  |  GEM low-rate science livetime counter live (fld)
```

```
|   |  Low-rate science prescaled counter presc (fld)
|   |  GEM low-rate science discarded counter disc (fld)
|   |  GEM low-rate science sent counter sent (fld)
|  Timestamp seconds temsec (fld)
|  Timestamp subseconds temsub (fld)
|  TEM Deadtime Register Bitfield temlrs[0-15] (bf)
|   |  TEM Deadtime Low-rate Science Counter deadtime (0-23)

typedef struct _LHK_Lrs0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* Timestamp seconds */
    unsigned int         gemsec;

    /* Timestamp subseconds */
    unsigned int         gemsub;

    /* GEM Low-Rate Science Counter */
    LHK_GEMLRS           gemlrs;

    /* Timestamp seconds */
    unsigned int         temsec;

    /* Timestamp subseconds */
    unsigned int         temsub;

    /* TEM Deadtime Register Bitfield */
    LHK_TEMDEADTIMEREG   temlrs[16];

} LHK_Lrs0_Tlm;
```

Fields:

`GEMLRS` **(10.5.5)**

Attribute(s):

Brief: "GEM Low-Rate Science Counter"

Instance(s): `gemlrs`

Contains 4 GEM low-rate science counter values.

`TEMDEADTIMEREG` **(10.6.29)**

Attribute(s):

Brief: "TEM Deadtime Register Bitfield"

Instance(s): `temlrs`

Describes the TEM dead time low-rate science counter register.

`TIMESEC` **(10.7.48)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `gemsec`, `temsec`

`TIMESUBSEC` **(10.7.49)**

Attribute(s):

Brief: "Timestamp subseconds"

Instance(s): `gemsub`, `temsub`

`TSP16` **(10.7.54)**

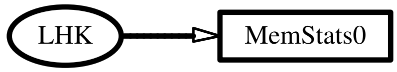
Attribute(s):

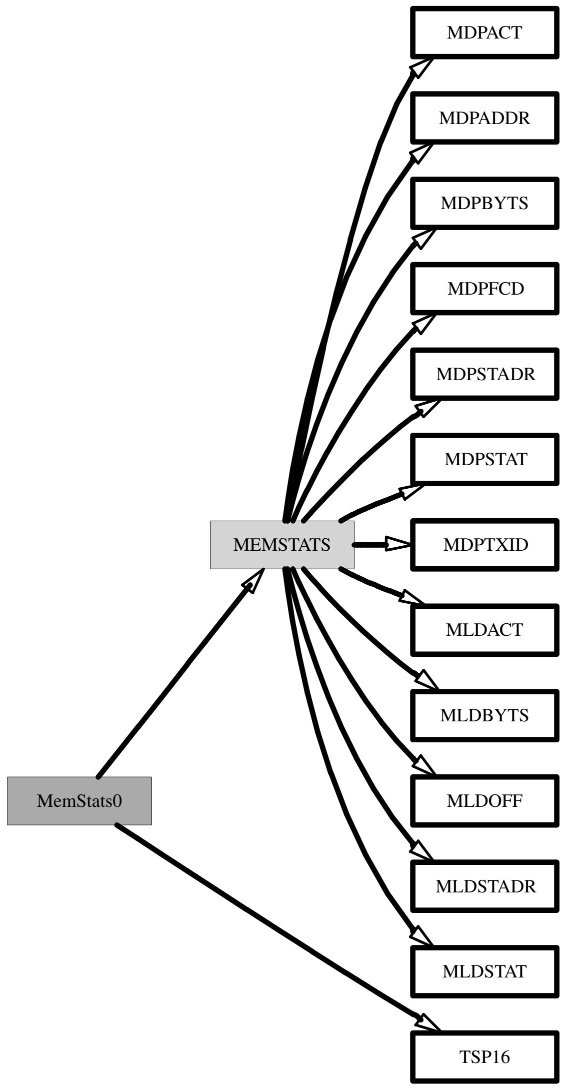
Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.36 `MemStats0` (556/0x22C)

Context:





Description:

"Memory Load/Dump Statistics" Telemetry Packet

Contains the statistics for memory loads and dumps on the SIU and EPU0.

Layout:

Memory Load/Dump Statistics (MemStats0) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x22C**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spare[0-2]															
014	**MEMSTATS:** mem[0-1]															

```
MemStats0 (pkt)
|   Spare 16 bit field spare[0-2] (fld)
|   MEM Statistics mem[0-1] (str)
|   |   Status of most recent load action ldstat (fld)
|   |   Memory load active flag ldact (fld)
|   |   Starting memory load address ldstart (fld)
|   |   Memory load total bytes ldbytes (fld)
|   |   Memory load offset ldoffset (fld)
|   |   Memory dump status dmpstat (fld)
|   |   Memory dump active dmpact (fld)
|   |   Memory dump start address dmpstart (fld)
|   |   Memory dump bytes dmpbytes (fld)
|   |   Memory dump address dmpaddr (fld)
|   |   Memory dump function code dmpfcode (fld)
|   |   Memory dump transaction ID dmptid (fld)
```

```
typedef struct _LHK_MemStats0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spare[3];

    /* MEM Statistics */
    LHK_MEMSTATS         mem[2];

} LHK_MemStats0_Tlm;
```

Fields:

MEMSTATS **(10.5.6)**

Attribute(s):

Brief: "MEM Statistics"

Instance(s): mem

Contains statistics for the MEM task.

`TSP16` **(10.7.54)**

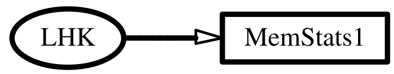
Attribute(s):

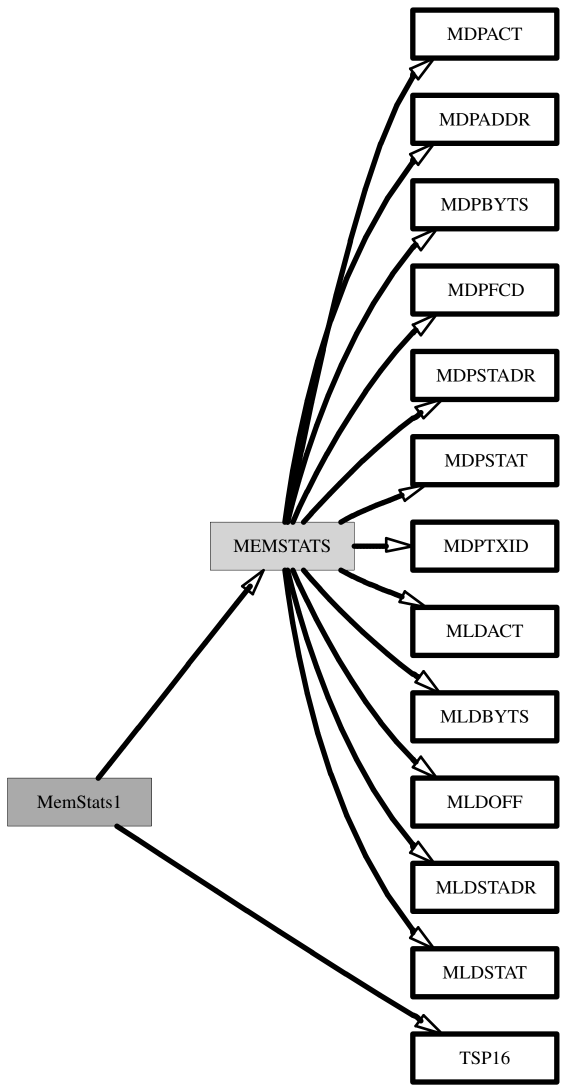
Brief: "Spare 16 bit field"

Instance(s): `spare`

10.4.37 `MemStats1` (557/0x22D)

Context:





Description:

"Memory Load/Dump Statistics" Telemetry Packet

Contains the statistics for memory loads and dumps on the EPU1 and EPU2.

Layout:

Memory Load/Dump Statistics (MemStats1) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x22D**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP16:** spares[0-2]															
014	**MEMSTATS:** mem[0-1]															

```
MemStats1 (pkt)
|  Spare 16 bit field spares[0-2] (fld)
|  MEM Statistics mem[0-1] (str)
|  |  Status of most recent load action ldstat (fld)
|  |  Memory load active flag ldact (fld)
|  |  Starting memory load address ldstart (fld)
|  |  Memory load total bytes ldbytes (fld)
|  |  Memory load offset ldoffset (fld)
|  |  Memory dump status dmpstat (fld)
|  |  Memory dump active dmpact (fld)
|  |  Memory dump start address dmpstart (fld)
|  |  Memory dump bytes dmpbytes (fld)
|  |  Memory dump address dmpaddr (fld)
|  |  Memory dump function code dmpfcode (fld)
|  |  Memory dump transaction ID dmptid (fld)
```

```
typedef struct _LHK_MemStats1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bit field */
    unsigned short       spares[3];

    /* MEM Statistics */
    LHK_MEMSTATS         mem[2];

} LHK_MemStats1_Tlm;
```

Fields:

MEMSTATS **(10.5.6)**

Attribute(s):

Brief: "MEM Statistics"

Instance(s): mem

Contains statistics for the MEM task.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `spares`

10.4.38 `PduEnv0` (542/0x21E)

Context:

Description:

"PDU Environmental Packet 0" Telemetry Packet

PDU Packet 0

Layout:

PDU Environmental Packet 0 (`PduEnv0`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x21E**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**PDUTEMPWRREG:** `tempwr`															
016	**PDUEPUCRATEPWR:** `epupwr`															
018	**ACDPWRREG:** `acdpwr`															
01A	**TEM33V:** `temvolt[0-15]`															
03A	**TEMPSTEMP:** `temps[0-15]`															
05A	**EPUV:** `epuvolt[0-2]`															
060	**EPUTEMP:** `eputemp[0-2]`															
066	**AEMDAQENV:** `dab`															
06E	**TSP8:** `pad[0-5]`															

```
PduEnv0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  PDU TEM Power Management Register tempwr (bf)
|  |  PDU TEM Power State temf (0) ... tem0 (15)
|  PDU EPU Crate Power States epupwr (bf)
|  |  Spare 10 bits spare0 (0-9)
|  |  PDU EPU Converter State epu2cvt (10) ... epu0cvt (12)
|  |  PDU EPU Crate Power State epu2pwr (13) ... epu0pwr (15)
|  PDU ADC Power Management Register acdpwr (bf)
|  |  Spare 13 bit field spare (0-12)
|  |  PDU ACD Power Converter Source cnvt (13)
```

```
|  |  PDU ACD Power Supply Source sup (14)
|  |  PDU ACD Power State acd (15)
|  TEM 3.3 Voltage ADC value temvolt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  TEM Power Supply Temperature ADC value temps[0-15] (same bf)
|  EPU Voltage ADC value epuvolt[0-2] (same bf)
|  EPU Temperature ADC value eputemp[0-2] (same bf)
|  AEM DAQ Environmental dab (str)
|  |  AEM FREE Board 3.3 Sum sumv (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  DAQ Board Temperature dabt (same bf)
|  |  AEM FREE Board 28V Sum sumi (same bf)
|  |  DAQ Board 3.3 Voltage dabv (same bf)
|  Spare byte field pad[0-5] (fld)

typedef struct _LHK_PduEnv0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* PDU TEM Power Management Register */
    LHK_PDUTEMPWRREG     tempwr;

    /* PDU EPU Crate Power States */
    LHK_PDUEPUCRATEPWR   epupwr;

    /* PDU ADC Power Management Register */
    LHK_ACDPWRREG        acdpwr;

    /* TEM 3.3 Voltage ADC value */
    LHK_TEM33V           temvolt[16];

    /* TEM Power Supply Temperature ADC value */
    LHK_TEMPSTEMP        temps[16];

    /* EPU Voltage ADC value */
    LHK_EPUV             epuvolt[3];

    /* EPU Temperature ADC value */
    LHK_EPUTEMP          eputemp[3];

    /* AEM DAQ Environmental */
    LHK_AEMDAQENV        dab;

    /* Spare byte field */
    unsigned char        pad[6];

} LHK_PduEnv0_Tlm;
```

Fields:

`ACDPWRREG` **(10.6.2)**

Attribute(s):

Brief: "PDU ADC Power Management Register"

Instance(s): `acdpwr`

`AEMDAQENV` **(10.5.0)**

Attribute(s):

Brief: "AEM DAQ Environmental"

Instance(s): `dab`

`EPUTEMP` **(10.6.19)**

Attribute(s):

Brief: "EPU Temperature ADC value"

Instance(s): `eputemp`

`EPUV` **(10.6.20)**

Attribute(s):

Brief: "EPU Voltage ADC value"

Instance(s): `epuvolt`

`PDUEPUCRATEPWR` **(10.6.23)**

Attribute(s):

Brief: "PDU EPU Crate Power States"

Instance(s): `epupwr`

`PDUTEMPWRREG` **(10.6.24)**

Attribute(s):

Brief: "PDU TEM Power Management Register"

Instance(s): `tempwr`

`TEM33V` **(10.6.28)**

Attribute(s):

Brief: "TEM 3.3 Voltage ADC value"

Instance(s): `temvolt`

Describes a TEM 3.3V ADC value and status.

`TEMPSTEMP` **(10.6.31)**

Attribute(s):

Brief: "TEM Power Supply Temperature ADC value"

Instance(s): `temps`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.39 `PduEnv1` (543/0x21F)

Context:

Description:

"PDU Environmental Packet 1" Telemetry Packet

PDU Packet 1

Layout:

PDU Environmental Packet 1 (PduEnv1) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x21F										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															

PDU Environmental Packet 1 (`PduEnv1`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** `spares[0-5]`															
014	**TEMPCBTEMP:** `pcbt[0-15]`															
034	**CALBSPTEMP:** `bspt[0-15]`															
054	**TSP8:** `pad[0-31]`															

```
PduEnv1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM PCB Temperature pcbt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  CAL Baseplate Temperature bspt[0-15] (same bf)
|  Spare byte field pad[0-31] (fld)
```

```
typedef struct _LHK_PduEnv1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM PCB Temperature */
    LHK_TEMPCBTEMP       pcbt[16];

    /* CAL Baseplate Temperature */
    LHK_CALBSPTEMP       bspt[16];

    /* Spare byte field */
    unsigned char        pad[32];

} LHK_PduEnv1_Tlm;
```

Fields:

`CALBSPTEMP` **(10.6.16)**

Attribute(s):

Brief: "CAL Baseplate Temperature"

Instance(s): `bspt`

`TEMPCBTEMP` **(10.6.30)**

Attribute(s):

Brief: "TEM PCB Temperature"

Instance(s): `pcbt`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.40 `PduEnv2` (544/0x220)

Context:

Description:

"PDU Environmental Packet 2" Telemetry Packet

PDU Packet 2

Layout:

PDU Environmental Packet 2 (PduEnv2) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x220**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**VCHPDSHPTEMP:** `vdi[0-11]`															
02C	**VCHPXLHPTEMP:** `vxi[0-11]`															
044	**VCHPRSVRHTRTEMP:** `vrh[0-11]`															
05C	**TSP8:** `pad[0-23]`															

```
PduEnv2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  VCHP-DSHP Interface Temperature vdi[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  VCHP-XLHP Interface Temperature vxi[0-11] (same bf)
|  VCHP Reservoir Heater Temperature vrh[0-11] (same bf)
|  Spare byte field pad[0-23] (fld)

typedef struct _LHK_PduEnv2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
```

```
    unsigned char        spares[6];

    /* VCHP-DSHP Interface Temperature */
    LHK_VCHPDSHPTEMP     vdi[12];

    /* VCHP-XLHP Interface Temperature */
    LHK_VCHPXLHPTEMP     vxi[12];

    /* VCHP Reservoir Heater Temperature */
    LHK_VCHPRSVRHTRTEMP  vrh[12];

    /* Spare byte field */
    unsigned char        pad[24];

} LHK_PduEnv2_Tlm;
```

Fields:

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

`VCHPDSHPTEMP` **(10.6.39)**

Attribute(s):

Brief: "VCHP-DSHP Interface Temperature"

Instance(s): `vdi`

`VCHPRSVRHTRTEMP` **(10.6.40)**

Attribute(s):

Brief: "VCHP Reservoir Heater Temperature"

Instance(s): `vrh`

`VCHPXLHPTEMP` **(10.6.41)**

Attribute(s):

Brief: "VCHP-XLHP Interface Temperature"

Instance(s): `vxi`

10.4.41 `PduEnv3` (545/0x221)

Context:

Description:

"PDU Environmental Packet 3" Telemetry Packet

PDU Packet 3

Layout:

PDU Environmental Packet 3 (PduEnv3) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x221										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**GRIDTEMP:** `gridt[0-11]`															
02C	**ACDSHELLTEMP:** `acdst[0-1]`															
030	**ACDPMTRAILTEMP:** `acdpr[0-3]`															
038	**ACDBEAGRIDTEMP:** `acdbg[0-1]`															
03C	**RADANHTRTEMP:** `rah[0-1]`															
040	**GRIDRADIFTEMP:** `grift[0-3]`															
048	**RADTEMP:** `rtemp[0-11]`															
060	**TSP8:** `pad[0-19]`															

```
PduEnv3 (pkt)
|  Spare byte field spares[0-5] (fld)
|  Grid Temperature gridt[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  ADC Shell Temperature acdst[0-1] (same bf)
|  ADC PMT Rail Temperature acdpr[0-3] (same bf)
|  ACD BEA Grid Interface Temperature acdbg[0-1] (same bf)
|  Radiator Anitfreeze Heater Temperature rah[0-1] (same bf)
|  Grid Radiator Interface Temperature grift[0-3] (same bf)
|  Radiator Temperature rtemp[0-11] (same bf)
|  Spare byte field pad[0-19] (fld)
```

```
typedef struct _LHK_PduEnv3_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* Grid Temperature */
```

```
    LHK_GRIDTEMP         gridt[12];

    /* ADC Shell Temperature */
    LHK_ACDSHELLTEMP     acdst[2];

    /* ADC PMT Rail Temperature */
    LHK_ACDPMTRAILTEMP   acdpr[4];

    /* ACD BEA Grid Interface Temperature */
    LHK_ACDBEAGRIDTEMP   acdbg[2];

    /* Radiator Anitfreeze Heater Temperature */
    LHK_RADANHTRTEMP     rah[2];

    /* Grid Radiator Interface Temperature */
    LHK_GRIDRADIFTEMP    grift[4];

    /* Radiator Temperature */
    LHK_RADTEMP          rtemp[12];

    /* Spare byte field */
    unsigned char        pad[20];

} LHK_PduEnv3_Tlm;
```

Fields:

`ACDBEAGRIDTEMP` **(10.6.0)**

Attribute(s):

Brief: "ACD BEA Grid Interface Temperature"

Instance(s): `acdbg`

`ACDPMTRAILTEMP` **(10.6.1)**

Attribute(s):

Brief: "ADC PMT Rail Temperature"

Instance(s): `acdpr`

`ACDSHELLTEMP` **(10.6.3)**

Attribute(s):

Brief: "ADC Shell Temperature"

Instance(s): `acdst`

`GRIDRADIFTEMP` **(10.6.21)**

Attribute(s):

Brief: "Grid Radiator Interface Temperature"

Instance(s): `grift`

`GRIDTEMP` **(10.6.22)**

Attribute(s):

Brief: "Grid Temperature"

Instance(s): `gridt`

`RADANHTRTEMP` **(10.6.25)**

Attribute(s):

Brief: "Radiator Anitfreeze Heater Temperature"

Instance(s): `rah`

`RADTEMP` **(10.6.26)**

Attribute(s):

Brief: "Radiator Temperature"

Instance(s): `rtemp`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.42 `PduEnv4` (546/0x222)

Context:

Description:

"PDU Environmental Packet 4" Telemetry Packet

PDU Packet 4

Layout:

PDU Environmental Packet 4 (`PduEnv4`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x222**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**PDUTEMPWRREG:** `tempwr`															
016	**PDUEPUCRATEPWR:** `epupwr`															
018	**ACDPWRREG:** `acdpwr`															
01A	**TEM33V:** `temvolt[0-15]`															
03A	**TEMPSTEMP:** `temps[0-15]`															
05A	**EPUV:** `epuvolt[0-2]`															
060	**EPUTEMP:** `eputemp[0-2]`															
066	**TSP8:** `pad[0-13]`															

```
PduEnv4 (pkt)
|  Spare byte field spares[0-5] (fld)
|  PDU TEM Power Management Register tempwr (bf)
|  |  PDU TEM Power State temf (0) ... tem0 (15)
|  PDU EPU Crate Power States epupwr (bf)
|  |  Spare 10 bits spare0 (0-9)
|  |  PDU EPU Converter State epu2cvt (10) ... epu0cvt (12)
|  |  PDU EPU Crate Power State epu2pwr (13) ... epu0pwr (15)
|  PDU ADC Power Management Register acdpwr (bf)
|  |  Spare 13 bit field spare (0-12)
|  |  PDU ACD Power Converter Source cnvt (13)
|  |  PDU ACD Power Supply Source sup (14)
|  |  PDU ACD Power State acd (15)
|  TEM 3.3 Voltage ADC value temvolt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  TEM Power Supply Temperature ADC value temps[0-15] (same bf)
|  EPU Voltage ADC value epuvolt[0-2] (same bf)
|  EPU Temperature ADC value eputemp[0-2] (same bf)
|  Spare byte field pad[0-13] (fld)
```

```
typedef struct _LHK_PduEnv4_Tlm {
```

```
    /* Telemetry header */
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* PDU TEM Power Management Register */
    LHK_PDUTEMPWRREG    tempwr;

    /* PDU EPU Crate Power States */
    LHK_PDUEPUCRATEPWR  epupwr;

    /* PDU ADC Power Management Register */
    LHK_ACDPWRREG       acdpwr;

    /* TEM 3.3 Voltage ADC value */
    LHK_TEM33V          temvolt[16];

    /* TEM Power Supply Temperature ADC value */
    LHK_TEMPSTEMP       temps[16];

    /* EPU Voltage ADC value */
    LHK_EPUV            epuvolt[3];

    /* EPU Temperature ADC value */
    LHK_EPUTEMP         eputemp[3];

    /* Spare byte field */
    unsigned char       pad[14];

} LHK_PduEnv4_Tlm;
```

Fields:

`ACDPWRREG` **(10.6.2)**

Attribute(s):

Brief: "PDU ADC Power Management Register"

Instance(s): `acdpwr`

`EPUTEMP` **(10.6.19)**

Attribute(s):

Brief: "EPU Temperature ADC value"

Instance(s): `eputemp`

`EPUV` **(10.6.20)**

Attribute(s):

Brief: "EPU Voltage ADC value"

Instance(s): `epuvolt`

`PDUEPUCRATEPWR` **(10.6.23)**

Attribute(s):

Brief: "PDU EPU Crate Power States"

Instance(s): `epupwr`

`PDUTEMPWRREG` **(10.6.24)**

Attribute(s):

Brief: "PDU TEM Power Management Register"

Instance(s): `tempwr`

`TEM33V` **(10.6.28)**

Attribute(s):

Brief: "TEM 3.3 Voltage ADC value"

Instance(s): `temvolt`

Describes a TEM 3.3V ADC value and status.

`TEMPSTEMP` **(10.6.31)**

Attribute(s):

Brief: "TEM Power Supply Temperature ADC value"

Instance(s): `temps`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.43 `PduEnv5` (547/0x223)

Context:

Description:

"PDU Environmental Packet 5" Telemetry Packet

PDU Packet 5

Layout:

PDU Environmental Packet 5 (PduEnv5) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x223										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**TEMPCBTEMP:** `pcbt[0-15]`															
034	**CALBSPTEMP:** `bspt[0-15]`															
054	**TSP8:** `pad[0-31]`															

```
PduEnv5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM PCB Temperature pcbt[0-15] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  CAL Baseplate Temperature bspt[0-15] (same bf)
|  Spare byte field pad[0-31] (fld)
```

```
typedef struct _LHK_PduEnv5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM PCB Temperature */
    LHK_TEMPCBTEMP       pcbt[16];

    /* CAL Baseplate Temperature */
    LHK_CALBSPTEMP       bspt[16];

    /* Spare byte field */
    unsigned char        pad[32];

} LHK_PduEnv5_Tlm;
```

Fields:

`CALBSPTEMP` **(10.6.16)**

Attribute(s):

Brief: "CAL Baseplate Temperature"

Instance(s): `bspt`

`TEMPCBTEMP` **(10.6.30)**

Attribute(s):

Brief: "TEM PCB Temperature"

Instance(s): `pcbt`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.44 `PduEnv6` (548/0x224)

Context:

Description:

"PDU Environmental Packet 6" Telemetry Packet

PDU Packet 6

Layout:

PDU Environmental Packet 6 (`PduEnv6`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x224**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**VCHPDSHPTEMP:** `vdi[0-11]`															

PDU Environmental Packet 6 (PduEnv6) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
02C	**VCHPXLHPTEMP:** vxi[0-11]															
044	**VCHPRSVRHTRTEMP:** vrh[0-11]															
05C	**TSP8:** pad[0-23]															

```
PduEnv6 (pkt)
|  Spare byte field spares[0-5] (fld)
|  VCHP-DSHP Interface Temperature vdi[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  VCHP-XLHP Interface Temperature vxi[0-11] (same bf)
|  VCHP Reservoir Heater Temperature vrh[0-11] (same bf)
|  Spare byte field pad[0-23] (fld)
```

```
typedef struct _LHK_PduEnv6_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* VCHP-DSHP Interface Temperature */
    LHK_VCHPDSHPTEMP     vdi[12];

    /* VCHP-XLHP Interface Temperature */
    LHK_VCHPXLHPTEMP     vxi[12];

    /* VCHP Reservoir Heater Temperature */
    LHK_VCHPRSVRHTRTEMP  vrh[12];

    /* Spare byte field */
    unsigned char        pad[24];

} LHK_PduEnv6_Tlm;
```

Fields:

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `pad`, `spares`

`VCHPDSHPTEMP` **(10.6.39)**

Attribute(s):

Brief: "VCHP-DSHP Interface Temperature"

Instance(s): `vdi`

`VCHPRSVRHTRTEMP` **(10.6.40)**

Attribute(s):

Brief: "VCHP Reservoir Heater Temperature"

Instance(s): `vrh`

`VCHPXLHPTEMP` **(10.6.41)**

Attribute(s):

Brief: "VCHP-XLHP Interface Temperature"

Instance(s): `vxi`

10.4.45 `PduEnv7` (549/0x225)

Context:

Description:

"PDU Environmental Packet 7" Telemetry Packet

PDU Packet 7

Layout:

PDU Environmental Packet 7 (`PduEnv7`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x225**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**GRIDTEMP:** `gridt[0-11]`															
02C	**ACDSHELLTEMP:** `acdst[0-1]`															
030	**ACDPMTRAILTEMP:** `acdpr[0-3]`															
038	**ACDBEAGRIDTEMP:** `acdbg[0-1]`															
03C	**RADANHTRTEMP:** `rah[0-1]`															
040	**GRIDRADIFTEMP:** `grift[0-3]`															
048	**RADTEMP:** `rtemp[0-11]`															

PDU Environmental Packet 7 (`PduEnv7`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
060	**TSP8:** `pad[0-19]`															

```
PduEnv7 (pkt)
|  Spare byte field spares[0-5] (fld)
|  Grid Temperature gridt[0-11] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  ADC Shell Temperature acdst[0-1] (same bf)
|  ADC PMT Rail Temperature acdpr[0-3] (same bf)
|  ACD BEA Grid Interface Temperature acdbg[0-1] (same bf)
|  Radiator Anitfreeze Heater Temperature rah[0-1] (same bf)
|  Grid Radiator Interface Temperature grift[0-3] (same bf)
|  Radiator Temperature rtemp[0-11] (same bf)
|  Spare byte field pad[0-19] (fld)
```

```
typedef struct _LHK_PduEnv7_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* Grid Temperature */
    LHK_GRIDTEMP        gridt[12];

    /* ADC Shell Temperature */
    LHK_ACDSHELLTEMP    acdst[2];

    /* ADC PMT Rail Temperature */
    LHK_ACDPMTRAILTEMP  acdpr[4];

    /* ACD BEA Grid Interface Temperature */
    LHK_ACDBEAGRIDTEMP  acdbg[2];

    /* Radiator Anitfreeze Heater Temperature */
    LHK_RADANHTRTEMP    rah[2];

    /* Grid Radiator Interface Temperature */
    LHK_GRIDRADIFTEMP   grift[4];

    /* Radiator Temperature */
    LHK_RADTEMP         rtemp[12];

    /* Spare byte field */
    unsigned char       pad[20];

} LHK_PduEnv7_Tlm;
```

Fields:

`ACDBEAGRIDTEMP` **(10.6.0)**

Attribute(s):

Brief: "ACD BEA Grid Interface Temperature"

Instance(s): `acdbg`

`ACDPMTRAILTEMP` **(10.6.1)**

Attribute(s):

Brief: "ADC PMT Rail Temperature"

Instance(s): `acdpr`

`ACDSHELLTEMP` **(10.6.3)**

Attribute(s):

Brief: "ADC Shell Temperature"

Instance(s): `acdst`

`GRIDRADIFTEMP` **(10.6.21)**

Attribute(s):

Brief: "Grid Radiator Interface Temperature"

Instance(s): `grift`

`GRIDTEMP` **(10.6.22)**

Attribute(s):

Brief: "Grid Temperature"

Instance(s): `gridt`

`RADANHTRTEMP` **(10.6.25)**

Attribute(s):

Brief: "Radiator Anitfreeze Heater Temperature"

Instance(s): `rah`

`RADTEMP` **(10.6.26)**

Attribute(s):

Brief: "Radiator Temperature"

Instance(s): `rtemp`

`TSP8` **(10.7.56)**

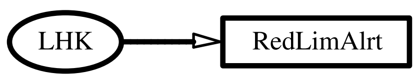
Attribute(s):

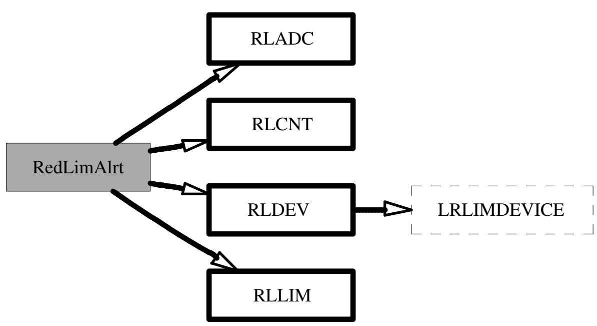
Brief: "Spare byte field"

Instance(s): `pad`, `spares`

10.4.46 `RedLimAlrt` (851/0x353)

Context:





Description:

"Red Limit Alert Packet" Telemetry Packet

This packet reports alert data related to an ADC limit threshold violation.

Layout:

Red Limit Alert Packet (`RedLimAlrt`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

Red Limit Alert Packet (RedLimAlrt) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x353**										
002	**SF**		**Sequence Count**													
004	**Packet Length=15**															
006	**Timestamp**															
00E	**RLDEV:** `dev`															
010	**RLADC:** `adc`															
012	**RLCNT:** `cnt`															
014	**RLLIM:** `lim`															

```
RedLimAlrt (pkt)
|  Red Limit Device Opcode dev (fld)
|  Red Limit ADC Number adc (fld)
|  Red Limit ADC Counts cnt (fld)
|  Red Limit Threshold lim (fld)
```

```
typedef struct _LHK_RedLimAlrt_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Red Limit Device Opcode */
    unsigned short       dev;

    /* Red Limit ADC Number */
    unsigned short       adc;

    /* Red Limit ADC Counts */
    unsigned short       cnt;

    /* Red Limit Threshold */
    unsigned short       lim;

} LHK_RedLimAlrt_Tlm;
```

Fields:

`RLADC` **(10.7.42)**

Attribute(s):

Brief: "Red Limit ADC Number"

Instance(s): `adc`

ACD number that exceeded its red limit.

`RLCNT` **(10.7.43)**

Attribute(s):

Brief: "Red Limit ADC Counts"

Instance(s): `cnt`

This field is used to report the raw ADC count value that exceeded the specified red limit.

`RLDEV` **(10.7.44)**

Attribute(s):
dsc: LRLIMDEVICE (10.9.8)

Brief: "Red Limit Device Opcode"

Instance(s): `dev`

Opcode specifying the device that exceeded a red limit.

`RLLIM` **(10.7.45)**

Attribute(s):

Brief: "Red Limit Threshold"

Instance(s): `lim`

This field reports the limit threshold that was used to evaluate the exceeded ADC value.

10.4.47 `TemEnvPwr0` (528/0x210)

Context:

Description:

"TEM Power Packet 0" Telemetry Packet

Contains power specific ADC values for TEMs 0, 1, and 2.

Layout:

TEM Power Packet 0 (`TemEnvPwr0`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x210**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															

TEM Power Packet 0 (TemEnvPwr0) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TMUX0:** t0m0															
024	**TMUX1:** t0m1															
034	**TMUX0:** t1m0															
044	**TMUX1:** t1m1															
054	**TMUX0:** t2m0															
064	**TMUX1:** t2m1															

```
TemEnvPwr0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t0m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t0m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t1m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t1m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
```

```
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t2m0 (str)
|   |  TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  TKR 2.5 Current ADC value adc1 (same bf)
|   |  TKR 1.5 Voltage ADC value adc2 (same bf)
|   |  TKR 1.5 Current ADC value adc3 (same bf)
|   |  TKR 2.5 Voltage ADC value adc4 (same bf)
|   |  TKR 2.5 Current ADC value adc5 (same bf)
|   |  TKR Bias Voltage ADC value adc6 (same bf)
|   |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t2m1 (str)
|   |  CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TemEnvPwr0_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t0m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t0m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t1m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t1m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t2m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t2m1;
```

```
} LHK_TemEnvPwr0_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t0m0`, `t1m0`, `t2m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t0m1`, `t1m1`, `t2m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.48 `TemEnvPwr1` (529/0x211)

Context:

Description:

"TEM Power Packet 1" Telemetry Packet

Contains power specific ADC values for TEMs 3,4, and 5.

Layout:

TEM Power Packet 1 (`TemEnvPwr1`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x211**										

TEM Power Packet 1 (TemEnvPwr1)
Telemetry Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	TSP8: spares[0-5]															
014	TMUX0: t3m0															
024	TMUX1: t3m1															
034	TMUX0: t4m0															
044	TMUX1: t4m1															
054	TMUX0: t5m0															
064	TMUX1: t5m1															

```
TemEnvPwr1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t3m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t3m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t4m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t4m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
```

```
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t5m0 (str)
|   |  TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  TKR 2.5 Current ADC value adc1 (same bf)
|   |  TKR 1.5 Voltage ADC value adc2 (same bf)
|   |  TKR 1.5 Current ADC value adc3 (same bf)
|   |  TKR 2.5 Voltage ADC value adc4 (same bf)
|   |  TKR 2.5 Current ADC value adc5 (same bf)
|   |  TKR Bias Voltage ADC value adc6 (same bf)
|   |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t5m1 (str)
|   |  CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TemEnvPwr1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t3m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t3m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t4m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t4m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t5m0;
```

```
    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t5m1;

} LHK_TemEnvPwr1_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t3m0`, `t4m0`, `t5m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t3m1`, `t4m1`, `t5m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.49 `TemEnvPwr2` (530/0x212)

Context:

Description:

"TEM Power Packet 2" Telemetry Packet

Contains power specific ADC values for TEMs 6, 7, and 8.

Layout:

TEM Power Packet 2 (`TemEnvPwr2`)
Telemetry Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	**Version=0**			**T=0**	**SH**	**APID=0x212**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `t6m0`															
024	**TMUX1:** `t6m1`															
034	**TMUX0:** `t7m0`															
044	**TMUX1:** `t7m1`															
054	**TMUX0:** `t8m0`															
064	**TMUX1:** `t8m1`															

```
TemEnvPwr2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t6m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t6m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t7m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t7m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
```

```
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs t8m0 (str)
|   |  TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  TKR 2.5 Current ADC value adc1 (same bf)
|   |  TKR 1.5 Voltage ADC value adc2 (same bf)
|   |  TKR 1.5 Current ADC value adc3 (same bf)
|   |  TKR 2.5 Voltage ADC value adc4 (same bf)
|   |  TKR 2.5 Current ADC value adc5 (same bf)
|   |  TKR Bias Voltage ADC value adc6 (same bf)
|   |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t8m1 (str)
|   |  CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TemEnvPwr2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t6m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t6m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t7m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t7m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t8m0;
```

```
    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t8m1;

} LHK_TemEnvPwr2_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t6m0`, `t7m0`, `t8m0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t6m1`, `t7m1`, `t8m1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.50 `TemEnvPwr3` (531/0x213)

Context:

Description:

"TEM Power Packet 3" Telemetry Packet

Contains power specific ADC values for TEMs 9, 10, and 11.

Layout:

TEM Power Packet 3 (TemEnvPwr3) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x213**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TMUX0:** t9m0															
024	**TMUX1:** t9m1															
034	**TMUX0:** tAm0															
044	**TMUX1:** tAm1															
054	**TMUX0:** tBm0															
064	**TMUX1:** tBm1															

```
TemEnvPwr3 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs t9m0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs t9m1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs tAm0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tAm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
```

```
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs tBm0 (str)
|   |  TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  TKR 2.5 Current ADC value adc1 (same bf)
|   |  TKR 1.5 Voltage ADC value adc2 (same bf)
|   |  TKR 1.5 Current ADC value adc3 (same bf)
|   |  TKR 2.5 Voltage ADC value adc4 (same bf)
|   |  TKR 2.5 Current ADC value adc5 (same bf)
|   |  TKR Bias Voltage ADC value adc6 (same bf)
|   |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tBm1 (str)
|   |  CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TemEnvPwr3_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            t9m0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            t9m1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tAm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tAm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tBm0;
```

```
    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1             tBm1;

} LHK_TemEnvPwr3_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `t9m0`, `tAm0`, `tBm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `t9m1`, `tAm1`, `tBm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.51 `TemEnvPwr4` (532/0x214)

Context:

Description:

"TEM Power Packet 4" Telemetry Packet

Contains power specific ADC values for TEMs 12, 13, and 14.

Layout:

TEM Power Packet 4 (TemEnvPwr4) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x214										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	TSP8: spares[0-5]															
014	TMUX0: tCm0															
024	TMUX1: tCm1															
034	TMUX0: tDm0															
044	TMUX1: tDm1															
054	TMUX0: tEm0															
064	TMUX1: tEm1															

```
TemEnvPwr4 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs tCm0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tCm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs tDm0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tDm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
```

```
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
|  TEM mux channel 0 environmental ADCs tEm0 (str)
|   |  TKR 2.5 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  TKR 2.5 Current ADC value adc1 (same bf)
|   |  TKR 1.5 Voltage ADC value adc2 (same bf)
|   |  TKR 1.5 Current ADC value adc3 (same bf)
|   |  TKR 2.5 Voltage ADC value adc4 (same bf)
|   |  TKR 2.5 Current ADC value adc5 (same bf)
|   |  TKR Bias Voltage ADC value adc6 (same bf)
|   |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tEm1 (str)
|   |  CAL 3.3 Voltage ADC value adc0 (bf)
|   |   |  ADC status bits stat (0-3)
|   |   |  ADC raw value val (4-15)
|   |  CAL 3.3 Current ADC value adc1 (same bf)
|   |  CAL 3.3 Voltage ADC value adc2 (same bf)
|   |  CAL 3.3 Current ADC value adc3 (same bf)
|   |  CAL Bias Voltage ADC value adc4 (same bf)
|   |  CAL Bias Current ADC value adc5 (same bf)
|   |  TEM 3.3 Voltage ADC value adc6 (same bf)
|   |  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TemEnvPwr4_Tlm {

    /* Telemetry header */
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0           tCm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1           tCm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0           tDm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1           tDm1;

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0           tEm0;
```

```
    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tEm1;

} LHK_TemEnvPwr4_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `tCm0`, `tDm0`, `tEm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `tCm1`, `tDm1`, `tEm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.52 `TemEnvPwr5` (533/0x215)

Context:

Description:

"TEM Power Packet 5" Telemetry Packet

Contains powr specific ADC values for TEM 15.

Layout:

TEM Power Packet 5 (TemEnvPwr5) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x215										
002	SF		Sequence Count													
004	Packet Length=109															
006	Timestamp															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX0:** `tFm0`															
024	**TMUX1:** `tFm1`															
034	**TSP16:** `pad[0-31]`															

```
TemEnvPwr5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 0 environmental ADCs tFm0 (str)
|  |  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  TKR 2.5 Current ADC value adc1 (same bf)
|  |  TKR 1.5 Voltage ADC value adc2 (same bf)
|  |  TKR 1.5 Current ADC value adc3 (same bf)
|  |  TKR 2.5 Voltage ADC value adc4 (same bf)
|  |  TKR 2.5 Current ADC value adc5 (same bf)
|  |  TKR Bias Voltage ADC value adc6 (same bf)
|  |  TKR Bias Current ADC value adc7 (same bf)
|  TEM mux channel 1 environmental ADCs tFm1 (str)
|  |  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  |  CAL 3.3 Current ADC value adc1 (same bf)
|  |  CAL 3.3 Voltage ADC value adc2 (same bf)
|  |  CAL 3.3 Current ADC value adc3 (same bf)
|  |  CAL Bias Voltage ADC value adc4 (same bf)
|  |  CAL Bias Current ADC value adc5 (same bf)
|  |  TEM 3.3 Voltage ADC value adc6 (same bf)
|  |  TEM 3.3 Current ADC value adc7 (same bf)
|  Spare 16 bit field pad[0-31] (fld)
```

```
typedef struct _LHK_TemEnvPwr5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 0 environmental ADCs */
    LHK_TMUX0            tFm0;

    /* TEM mux channel 1 environmental ADCs */
    LHK_TMUX1            tFm1;
```

```
    /* Spare 16 bit field */
    unsigned short       pad[32];

} LHK_TemEnvPwr5_Tlm;
```

Fields:

`TMUX0` **(10.5.8)**

Attribute(s):

Brief: "TEM mux channel 0 environmental ADCs"

Instance(s): `tFm0`

Describes 8 ADC quantities for TEM mux channel 0.

`TMUX1` **(10.5.9)**

Attribute(s):

Brief: "TEM mux channel 1 environmental ADCs"

Instance(s): `tFm1`

Contains 6 CAL and 2 TEM power ADC environmental values.

`TSP16` **(10.7.54)**

Attribute(s):

Brief: "Spare 16 bit field"

Instance(s): `pad`

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.53 `TemEnvTemp0` (534/0x216)

Context:

Description:

"TEM Temperature Packet 0" Telemetry Packet

Contains temperature specific ADC values for TEMs 0 and 1.

Layout:

TEM Temperature Packet 0 (TemEnvTemp0) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x216**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** t0m2															
024	**TMUX3:** t0m3															
034	**TMUX4:** t0m4															
044	**TMUX2:** t1m0															
054	**TMUX3:** t1m1															
064	**TMUX4:** t1m2															

```
TemEnvTemp0 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t0m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t0m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t0m4 (same str)
|  TEM mux channel 2 environmental ADCs t1m0 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t1m1 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t1m2 (same str)
```

```
typedef struct _LHK_TemEnvTemp0_Tlm {

    /* Telemetry header */
```

```
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t0m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t0m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t0m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t1m0;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t1m1;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t1m2;

} LHK_TemEnvTemp0_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t0m2`, `t1m0`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t0m3`, `t1m1`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t0m4`, `t1m2`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.54 `TemEnvTemp1` (535/0x217)

Context:

Description:

"TEM Temperature Packet 1" Telemetry Packet

Contains temperature specific ADC values for TEMs 2 and 3.

Layout:

TEM Temperature Packet 1 (`TemEnvTemp1`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x217**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t2m2`															
024	**TMUX3:** `t2m3`															
034	**TMUX4:** `t2m4`															
044	**TMUX2:** `t3m2`															
054	**TMUX3:** `t3m3`															
064	**TMUX4:** `t3m4`															

```
TemEnvTemp1 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t2m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t2m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
```

```
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t2m4 (same str)
|  TEM mux channel 2 environmental ADCs t3m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t3m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t3m4 (same str)

typedef struct _LHK_TemEnvTemp1_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t2m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t2m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t2m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t3m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t3m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t3m4;

} LHK_TemEnvTemp1_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t2m2`, `t3m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t2m3`, `t3m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t2m4`, `t3m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.55 `TemEnvTemp2` (536/0x218)

Context:

Description:

"TEM Temperature Packet 2" Telemetry Packet

Contains temperature specific ADC values for TEMs 4 and 5.

Layout:

TEM Temperature Packet 2 (`TemEnvTemp2`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x218**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															

TEM Temperature Packet 2 (TemEnvTemp2) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** t4m2															
024	**TMUX3:** t4m3															
034	**TMUX4:** t4m4															
044	**TMUX2:** t5m2															
054	**TMUX3:** t5m3															
064	**TMUX4:** t5m4															

```
TemEnvTemp2 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t4m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t4m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t4m4 (same str)
|  TEM mux channel 2 environmental ADCs t5m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t5m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t5m4 (same str)
```

```
typedef struct _LHK_TemEnvTemp2_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t4m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t4m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t4m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t5m2;

    /* TEM mux channel 3 environmental ADCs */
```

```
    LHK_TMUX3            t5m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t5m4;

} LHK_TemEnvTemp2_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t4m2`, `t5m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t4m3`, `t5m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t4m4`, `t5m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.56 `TemEnvTemp3` (537/0x219)

Context:

Description:

"TEM Temperature Packet 3" Telemetry Packet

Contains temperature specific ADC values for TEMs 6 and 7.

Layout:

TEM Temperature Packet 3 (TemEnvTemp3) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x219**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** t6m2															
024	**TMUX3:** t6m3															
034	**TMUX4:** t6m4															
044	**TMUX2:** t7m2															
054	**TMUX3:** t7m3															
064	**TMUX4:** t7m4															

```
TemEnvTemp3 (pkt)
│  Spare byte field spares[0-5] (fld)
│  TEM mux channel 2 environmental ADCs t6m2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs t6m3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs t6m4 (same str)
│  TEM mux channel 2 environmental ADCs t7m2 (str)
│  │  CAL AFFE temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 3 environmental ADCs t7m3 (str)
│  │  TKR Cable Temperature ADC value adcs[0-7] (bf)
│  │  │  ADC status bits stat (0-3)
│  │  │  ADC raw value val (4-15)
│  TEM mux channel 4 environmental ADCs t7m4 (same str)
```

```
typedef struct _LHK_TemEnvTemp3_Tlm {

    /* Telemetry header */
```

```
    char                hdr[14];

    /* Spare byte field */
    unsigned char       spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t6m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t6m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t6m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2           t7m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3           t7m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4           t7m4;

} LHK_TemEnvTemp3_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t6m2`, `t7m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t6m3`, `t7m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t6m4`, `t7m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.57 `TemEnvTemp4` (538/0x21A)

Context:

Description:

"TEM Temperature Packet 4" Telemetry Packet

Contains temperature specific ADC values for TEMs 8 and 9.

Layout:

TEM Temperature Packet 4 (TemEnvTemp4) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x21A**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `t8m2`															
024	**TMUX3:** `t8m3`															
034	**TMUX4:** `t8m4`															
044	**TMUX2:** `t9m2`															
054	**TMUX3:** `t9m3`															
064	**TMUX4:** `t9m4`															

```
TemEnvTemp4 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs t8m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t8m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
```

```
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t8m4 (same str)
|  TEM mux channel 2 environmental ADCs t9m2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs t9m3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs t9m4 (same str)
```

```
typedef struct _LHK_TemEnvTemp4_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t8m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t8m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t8m4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            t9m2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            t9m3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            t9m4;

} LHK_TemEnvTemp4_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `t8m2`, `t9m2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `t8m3`, `t9m3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `t8m4`, `t9m4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.58 `TemEnvTemp5` (539/0x21B)

Context:

Description:

"TEM Temperature Packet 5" Telemetry Packet

Contains temperature specific ADC values for TEMs A and B.

Layout:

TEM Temperature Packet 5 (`TemEnvTemp5`) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x21B**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															

TEM Temperature Packet 5 (TemEnvTemp5) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
00E	**TSP8:** spares[0-5]															
014	**TMUX2:** tAm2															
024	**TMUX3:** tAm3															
034	**TMUX4:** tAm4															
044	**TMUX2:** tBm2															
054	**TMUX3:** tBm3															
064	**TMUX4:** tBm4															

```
TemEnvTemp5 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs tAm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tAm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tAm4 (same str)
|  TEM mux channel 2 environmental ADCs tBm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tBm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tBm4 (same str)
```

```
typedef struct _LHK_TemEnvTemp5_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tAm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tAm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tAm4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tBm2;

    /* TEM mux channel 3 environmental ADCs */
```

```
    LHK_TMUX3            tBm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tBm4;

} LHK_TemEnvTemp5_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tAm2`, `tBm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tAm3`, `tBm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tAm4`, `tBm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.59 `TemEnvTemp6` (540/0x21C)

Context:

Description:

"TEM Temperature Packet 6" Telemetry Packet

Contains temperature specific ADC values for TEMs C and D.

Layout:

<table>
<tr><th colspan="17">TEM Temperature Packet 6 (TemEnvTemp6)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x21C</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=109</td></tr>
<tr><td>006</td><td colspan="16">Timestamp</td></tr>
<tr><td>00E</td><td colspan="16">TSP8: spares[0-5]</td></tr>
<tr><td>014</td><td colspan="16">TMUX2: tCm2</td></tr>
<tr><td>024</td><td colspan="16">TMUX3: tCm3</td></tr>
<tr><td>034</td><td colspan="16">TMUX4: tCm4</td></tr>
<tr><td>044</td><td colspan="16">TMUX2: tDm2</td></tr>
<tr><td>054</td><td colspan="16">TMUX3: tDm3</td></tr>
<tr><td>064</td><td colspan="16">TMUX4: tDm4</td></tr>
</table>

```
TemEnvTemp6 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs tCm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tCm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tCm4 (same str)
|  TEM mux channel 2 environmental ADCs tDm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tDm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tDm4 (same str)
```

```
typedef struct _LHK_TemEnvTemp6_Tlm {

    /* Telemetry header */
```

```
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tCm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tCm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tCm4;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tDm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tDm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tDm4;

} LHK_TemEnvTemp6_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tCm2`, `tDm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tCm3`, `tDm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tCm4`, `tDm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.4.60 `TemEnvTemp7` (541/0x21D)

Context:

Description:

"TEM Temperature Packet 7" Telemetry Packet

Contains temperature specific ADC values for TEMs E and F.

Layout:

TEM Temperature Packet 7 (`TemEnvTemp7`) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x21D**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**TSP8:** `spares[0-5]`															
014	**TMUX2:** `tEm3`															
024	**TMUX3:** `tEm4`															
034	**TMUX4:** `tEm5`															
044	**TMUX2:** `tFm2`															
054	**TMUX3:** `tFm3`															
064	**TMUX4:** `tFm4`															

```
TemEnvTemp7 (pkt)
|  Spare byte field spares[0-5] (fld)
|  TEM mux channel 2 environmental ADCs tEm3 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tEm4 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
```

```
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tEm5 (same str)
|  TEM mux channel 2 environmental ADCs tFm2 (str)
|  |  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 3 environmental ADCs tFm3 (str)
|  |  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  |  ADC status bits stat (0-3)
|  |  |  ADC raw value val (4-15)
|  TEM mux channel 4 environmental ADCs tFm4 (same str)
```

```
typedef struct _LHK_TemEnvTemp7_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare byte field */
    unsigned char        spares[6];

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tEm3;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tEm4;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tEm5;

    /* TEM mux channel 2 environmental ADCs */
    LHK_TMUX2            tFm2;

    /* TEM mux channel 3 environmental ADCs */
    LHK_TMUX3            tFm3;

    /* TEM mux channel 4 environmental ADCs */
    LHK_TMUX4            tFm4;

} LHK_TemEnvTemp7_Tlm;
```

Fields:

`TMUX2` **(10.5.10)**

Attribute(s):

Brief: "TEM mux channel 2 environmental ADCs"

Instance(s): `tEm3`, `tFm2`

Contains 8 AFFE ADC temperature values.

`TMUX3` **(10.5.11)**

Attribute(s):

Brief: "TEM mux channel 3 environmental ADCs"

Instance(s): `tEm4`, `tFm3`

Contains 8 TKR cable temperature ADC values.

`TMUX4` **(10.5.12)**

Attribute(s):

Brief: "TEM mux channel 4 environmental ADCs"

Instance(s): `tEm5`, `tFm4`

Contains 8 TKR cable temperature ADC value and status.

`TSP8` **(10.7.56)**

Attribute(s):

Brief: "Spare byte field"

Instance(s): `spares`

10.5 Telemetry Structs

10.5.0 **AEMDAQENV** (AEM DAQ Environmental) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 64 bits (8 bytes)

Description:

AEM DAQ Environmental (**AEMDAQENV**) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**AEMFR33ISUM:** `sumv`															
002	**DABTEMP:** `dabt`															
004	**AEMFR28ISUM:** `sumi`															
006	**DAQ33V:** `dabv`															

```
AEMDAQENV (str)
|  AEM FREE Board 3.3 Sum sumv (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  DAQ Board Temperature dabt (same bf)
|  AEM FREE Board 28V Sum sumi (same bf)
|  DAQ Board 3.3 Voltage dabv (same bf)
```

```
typedef struct _LHK_AEMDAQENV {

    /* AEM FREE Board 3.3 Sum */
    LHK_AEMFR33ISUM      sumv;

    /* DAQ Board Temperature */
    LHK_DABTEMP          dabt;

    /* AEM FREE Board 28V Sum */
    LHK_AEMFR28ISUM      sumi;

    /* DAQ Board 3.3 Voltage */
    LHK_DAQ33V           dabv;

} LHK_AEMDAQENV;
```

Fields:

`AEMFR28ISUM` **(10.6.4)**

Attribute(s):

Brief: "AEM FREE Board 28V Sum"

Instance(s): `sumi`

The sum of the high voltage (28V) currents measured over all FREE boards.

`AEMFR33ISUM` **(10.6.5)**

Attribute(s):

Brief: "AEM FREE Board 3.3 Sum"

Instance(s): `sumv`

The sum of the digital(3.3) currents of all FREE boards.

`DABTEMP` **(10.6.17)**

Attribute(s):

Brief: "DAQ Board Temperature"

Instance(s): `dabt`

The temperature of the DAQ board.

`DAQ33V` **(10.6.18)**

Attribute(s):

Brief: "DAQ Board 3.3 Voltage"

Instance(s): `dabv`

The digital (3.3) voltage of the DAQ board.

Used by:

`DiagPduEnv0 (10.4.12)`

10.5.1 `AEMFRENV` (AEM Free Board Environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 64 bits (8 bytes)

Description:

AEM Free Board Environmental ADCs (**AEMFRENV**) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**AEMFRVDD:** vdd															
002	**AEMFRTEMP:** temp															
004	**AEMFRHV1:** hv1															
006	**AEMFRHV2:** hv2															

```
AEMFRENV (str)
|  AEM VDD vdd (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  AEM Free board Temperature temp (same bf)
|  AEM Free Board HV1 hv1 (same bf)
|  AEM Free Board HV2 hv2 (same bf)
```

```
typedef struct _LHK_AEMFRENV {

    /* AEM VDD */
    LHK_AEMFRVDD         vdd;

    /* AEM Free board Temperature */
    LHK_AEMFRTEMP        temp;

    /* AEM Free Board HV1 */
    LHK_AEMFRHV1         hv1;

    /* AEM Free Board HV2 */
    LHK_AEMFRHV2         hv2;

} LHK_AEMFRENV;
```

Fields:

AEMFRHV1 **(10.6.6)**

Attribute(s):

Brief: "AEM Free Board HV1"

Instance(s): hv1

AEMFRHV2 **(10.6.7)**

Attribute(s):

Brief: "AEM Free Board HV2"

Instance(s): hv2

`AEMFRTEMP` **(10.6.9)**

Attribute(s):

Brief: "AEM Free board Temperature"

Instance(s): `temp`

`AEMFRVDD` **(10.6.10)**

Attribute(s):

Brief: "AEM VDD"

Instance(s): `vdd`

Used by:

`AemEnv0 (10.4.0)`

10.5.2 `CMDCNTRS` (Command Counters) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 128 bits (16 bytes)

Description:

Structure to hold per task command count information.

Command Counters (CMDCNTRS) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	TIMESEC: `sec`															
004	TIMESUBSEC: `subsec`															
008	CMDCNTS: `sent`															
00C	CMDDISPF: `dispf`															
00E	CMDEXEF: `exef`															

```
CMDCNTRS (str)
|  Timestamp seconds sec (fld)
|  Timestamp subseconds subsec (fld)
|  Commands sent counter sent (fld)
|  Command dispatch failure counter dispf (fld)
|  Command execution failure counter exef (fld)
```

```
typedef struct _LHK_CMDCNTRS {

    /* Timestamp seconds */
    unsigned int         sec;
```

```
    /* Timestamp subseconds */
    unsigned int         subsec;

    /* Commands sent counter */
    unsigned int         sent;

    /* Command dispatch failure counter */
    unsigned short       dispf;

    /* Command execution failure counter */
    unsigned short       exef;

} LHK_CMDCNTRS;
```

Fields:

`CMDCNTS` **(10.7.3)**

Attribute(s):

Brief: "Commands sent counter"

Instance(s): `sent`

Commands sent to task.

`CMDDISPF` **(10.7.4)**

Attribute(s):

Brief: "Command dispatch failure counter"

Instance(s): `dispf`

Commands failing task dispatch.

`CMDEXEF` **(10.7.5)**

Attribute(s):

Brief: "Command execution failure counter"

Instance(s): `exef`

Commands failing task execution

`TIMESEC` **(10.7.48)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `sec`

`TIMESUBSEC` **(10.7.49)**

Attribute(s):

Brief: "Timestamp subseconds"

Instance(s): `subsec`

Used by:

`CmdCnt0 (10.4.1)`

10.5.3 `CPUMETR` (CPU Metrics) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 16 bits (2 bytes)

Description:

Structure containing metrics for a single CPU.

CPU Metrics (CPUMETR) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	CPUJT: `temp`															

```
CPUMETR (str)
|  CPU Junction Temperature temp (fld)
```

```
typedef struct _LHK_CPUMETR {

    /* CPU Junction Temperature */
    unsigned short       temp;

} LHK_CPUMETR;
```

Fields:

`CPUJT` **(10.7.6)**

Attribute(s):

Brief: "CPU Junction Temperature"

Instance(s): `temp`

The CPU junction temperature, in degress Celcius.

Used by:

`CpuMetr (10.4.3)`

10.5.4 `FILESTATS` (FILE Upload Statistics) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 192 bits (24 bytes)

Description:

Describes the FILE upload current state.

FILE Upload Statistics (FILESTATS) Telemetry Field Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	FILESTATE: `state`															
004	FILESIZECUR: `cursize`															
008	FILEPKTCNT: `pktcnt`															
00C	FILERRCODE: `errcode`															
010	FILERRCNT: `errcnt`															
014	FILEIDCOM: `comid`															

```
FILESTATS (str)
|  FILE Upload State state (fld)
|  File Size Current cursize (fld)
|  FILE Packet Count pktcnt (fld)
|  FILE Error Code errcode (fld)
|  FILE Error Count errcnt (fld)
|  FILE ID Commit comid (fld)
```

```
typedef struct _LHK_FILESTATS {

    /* FILE Upload State */
    unsigned int         state;

    /* File Size Current */
    unsigned int         cursize;

    /* FILE Packet Count */
    unsigned int         pktcnt;

    /* FILE Error Code */
    unsigned int         errcode;

    /* FILE Error Count */
    unsigned int         errcnt;
```

```
    /* FILE ID Commit */
    unsigned int         comid;

} LHK_FILESTATS;
```

Fields:

`FILEIDCOM` **(10.7.11)**

Attribute(s):

Brief: "FILE ID Commit"

Instance(s): `comid`

The file ID value for commiting the file data to storage.

`FILEPKTCNT` **(10.7.12)**

Attribute(s):

Brief: "FILE Packet Count"

Instance(s): `pktcnt`

The number of vaild data packets received for the current file upload.

`FILERRCNT` **(10.7.13)**

Attribute(s):

Brief: "FILE Error Count"

Instance(s): `errcnt`

A count of the number of errors reported for the current file upload.

`FILERRCODE` **(10.7.14)**

Attribute(s):

Brief: "FILE Error Code"

Instance(s): `errcode`

The current file upload error indicator.

`FILESIZECUR` **(10.7.15)**

Attribute(s):

Brief: "File Size Current"

Instance(s): `cursize`

The number of data bytes received for the current file upload.

`FILESTATE` **(10.7.16)**

Attribute(s):

Brief: "FILE Upload State"

Instance(s): `state`

The current file upload state.

Used by:

`DiagFileStats (10.4.8)`

10.5.5 `GEMLRS` (GEM Low-Rate Science Counter) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 128 bits (16 bytes)

Description:

Contains 4 GEM low-rate science counter values.

GEM Low-Rate Science Counter (GEMLRS) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	GEMLRSLIVE: `live`															
004	GEMPRSCL: `presc`															
008	GEMLRSDISC: `disc`															
00C	GEMLRSSENT: `sent`															

```
GEMLRS (str)
|  GEM low-rate science livetime counter live (fld)
|  Low-rate science prescaled counter presc (fld)
|  GEM low-rate science discarded counter disc (fld)
|  GEM low-rate science sent counter sent (fld)
```

```
typedef struct _LHK_GEMLRS {

    /* GEM low-rate science livetime counter */
```

```
    unsigned int         live;

    /* Low-rate science prescaled counter */
    unsigned int         presc;

    /* GEM low-rate science discarded counter */
    unsigned int         disc;

    /* GEM low-rate science sent counter */
    unsigned int         sent;

} LHK_GEMLRS;
```

Fields:

`GEMLRSDISC` **(10.7.17)**

Attribute(s):

Brief: "GEM low-rate science discarded counter"

Instance(s): `disc`

Window turns discarded due to busy

`GEMLRSLIVE` **(10.7.18)**

Attribute(s):

Brief: "GEM low-rate science livetime counter"

Instance(s): `live`

1/deadtime statistics

`GEMLRSSENT` **(10.7.19)**

Attribute(s):

Brief: "GEM low-rate science sent counter"

Instance(s): `sent`

Window turns which result in a sent message

`GEMPRSCL` **(10.7.20)**

Attribute(s):

Brief: "Low-rate science prescaled counter"

Instance(s): `presc`

Window turns discarded due to prescaling

Used by:

`DiagLrs0 (10.4.9)`

10.5.6 **`MEMSTATS`** (MEM Statistics) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 384 bits (48 bytes)

Description:

Contains statistics for the MEM task.

MEM Statistics (MEMSTATS) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	MLDSTAT: `ldstat`															
004	MLDACT: `ldact`															
008	MLDSTADR: `ldstart`															
00C	MLDBYTS: `ldbytes`															
010	MLDOFF: `ldoffset`															
014	MDPSTAT: `dmpstat`															
018	MDPACT: `dmpact`															
01C	MDPSTADR: `dmpstart`															
020	MDPBYTS: `dmpbytes`															
024	MDPADDR: `dmpaddr`															
028	MDPFCD: `dmpfcode`															
02C	MDPTXID: `dmptid`															

```
MEMSTATS (str)
|  Status of most recent load action ldstat (fld)
|  Memory load active flag ldact (fld)
|  Starting memory load address ldstart (fld)
|  Memory load total bytes ldbytes (fld)
|  Memory load offset ldoffset (fld)
|  Memory dump status dmpstat (fld)
|  Memory dump active dmpact (fld)
|  Memory dump start address dmpstart (fld)
|  Memory dump bytes dmpbytes (fld)
|  Memory dump address dmpaddr (fld)
|  Memory dump function code dmpfcode (fld)
|  Memory dump transaction ID dmptid (fld)
```

```
typedef struct _LHK_MEMSTATS {
```

```
    /* Status of most recent load action */
    unsigned int         ldstat;

    /* Memory load active flag */
    unsigned int         ldact;

    /* Starting memory load address */
    unsigned int         ldstart;

    /* Memory load total bytes */
    unsigned int         ldbytes;

    /* Memory load offset */
    unsigned int         ldoffset;

    /* Memory dump status */
    unsigned int         dmpstat;

    /* Memory dump active */
    unsigned int         dmpact;

    /* Memory dump start address */
    unsigned int         dmpstart;

    /* Memory dump bytes */
    unsigned int         dmpbytes;

    /* Memory dump address */
    unsigned int         dmpaddr;

    /* Memory dump function code */
    unsigned int         dmpfcode;

    /* Memory dump transaction ID */
    unsigned int         dmptid;

} LHK_MEMSTATS;
```

Fields:

`MDPACT` **(10.7.24)**

Attribute(s):

Brief: "Memory dump active"

Instance(s): `dmpact`

Nonzero if dump is active.

`MDPADDR` **(10.7.25)**

Attribute(s):

Brief: "Memory dump address"

Instance(s): `dmpaddr`

Current dump address.

`MDPBYTS` **(10.7.26)**

Attribute(s):

Brief: "Memory dump bytes"

Instance(s): `dmpbytes`

Total number of bytes to dump.

`MDPFCD` **(10.7.27)**

Attribute(s):

Brief: "Memory dump function code"

Instance(s): `dmpfcode`

Function code for current dump.

`MDPSTADR` **(10.7.28)**

Attribute(s):

Brief: "Memory dump start address"

Instance(s): `dmpstart`

Starting dump address.

`MDPSTAT` **(10.7.29)**

Attribute(s):

Brief: "Memory dump status"

Instance(s): `dmpstat`

Status of most recent dump action.

`MDPTXID` **(10.7.30)**

Attribute(s):

Brief: "Memory dump transaction ID"

Instance(s): `dmptid`

Transaction ID for current dump.

`MLDACT` **(10.7.31)**

Attribute(s):

Brief: "Memory load active flag"

Instance(s): `ldact`

Nonzero if load is active

`MLDBYTS` **(10.7.32)**

Attribute(s):

Brief: "Memory load total bytes"

Instance(s): `ldbytes`

Total number of bytes to load.

`MLDOFF` **(10.7.33)**

Attribute(s):

Brief: "Memory load offset"

Instance(s): `ldoffset`

Current load offset.

`MLDSTADR` **(10.7.34)**

Attribute(s):

Brief: "Starting memory load address"

Instance(s): `ldstart`

`MLDSTAT` **(10.7.35)**

Attribute(s):

Brief: "Status of most recent load action"

Instance(s): `ldstat`

Used by:

`DiagMemStats0 (10.4.10)`

10.5.7 `RTSTATS` (1553 Remote Terminal Statistics) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 320 bits (40 bytes)

Description:

Contains the 1553 RT driver statistics.

1553 Remote Terminal Statistics (RTSTATS) Telemetry Field Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	RTERR: err															
004	INTRCNT: intr															
008	CRXPCNT: cxp															
00C	CRXBCNT: cxb															
010	CTXPCNT: txp															
014	CTXBCNT: txb															
018	HKPCNT: hkp															
01C	HKBCNT: hkb															
020	TLMPCNT: tlmp															
024	TLMBCNT: tlmb															

```
RTSTATS (str)
|  1553 Error Count err (fld)
|  1553 Interrupts intr (fld)
|  1553 Cmd Rx Packet Count cxp (fld)
|  1553 Cmd Rx Byte Count cxb (fld)
|  1553 Cmd Tx Packet Count txp (fld)
|  1553 Cmd Tx Byte Count txb (fld)
|  1553 Housekeeping Packet Count hkp (fld)
|  1553 Housekeeping Byte Count hkb (fld)
|  1553 Telemetry Packet Count tlmp (fld)
|  1553 Telemertry Byte Count tlmb (fld)
```

```
typedef struct _LHK_RTSTATS {

    /* 1553 Error Count */
    unsigned int         err;
```

```
    /* 1553 Interrupts */
    unsigned int         intr;

    /* 1553 Cmd Rx Packet Count */
    unsigned int         cxp;

    /* 1553 Cmd Rx Byte Count */
    unsigned int         cxb;

    /* 1553 Cmd Tx Packet Count */
    unsigned int         txp;

    /* 1553 Cmd Tx Byte Count */
    unsigned int         txb;

    /* 1553 Housekeeping Packet Count */
    unsigned int         hkp;

    /* 1553 Housekeeping Byte Count */
    unsigned int         hkb;

    /* 1553 Telemetry Packet Count */
    unsigned int         tlmp;

    /* 1553 Telemertry Byte Count */
    unsigned int         tlmb;

} LHK_RTSTATS;
```

Fields:

`CRXBCNT` **(10.7.7)**

Attribute(s):

Brief: "1553 Cmd Rx Byte Count"

Instance(s): `cxb`

A count of the number of CmdRx bytes received by the remote terminal

`CRXPCNT` **(10.7.8)**

Attribute(s):

Brief: "1553 Cmd Rx Packet Count"

Instance(s): `cxp`

A count of the number of CmdRx packets received by the remote terminal

`CTXBCNT` **(10.7.9)**

Attribute(s):

Brief: "1553 Cmd Tx Byte Count"

Instance(s): `txb`

A count of the number of CmdTx bytes sent by the remote terminal.

`CTXPCNT` **(10.7.10)**

Attribute(s):

Brief: "1553 Cmd Tx Packet Count"

Instance(s): `txp`

A count of the number of CmdTx packets sent by the remote terminal

`HKBCNT` **(10.7.21)**

Attribute(s):

Brief: "1553 Housekeeping Byte Count"

Instance(s): `hkb`

A count of the number of HKP bytes sent by the remote terminal

`HKPCNT` **(10.7.22)**

Attribute(s):

Brief: "1553 Housekeeping Packet Count"

Instance(s): `hkp`

A count of the number of HKP packets sent by the remote terminal

`INTRCNT` **(10.7.23)**

Attribute(s):

Brief: "1553 Interrupts"

Instance(s): `intr`

A count of the number of remote terminal device interrupts.

`RTERR` **(10.7.46)**

Attribute(s):

Brief: "1553 Error Count"

Instance(s): `err`

A count of the number of remote terminal errors.

`TLMBCNT` **(10.7.50)**

Attribute(s):

Brief: "1553 Telemertry Byte Count"

Instance(s): `tlmb`

A count of the number of Telem bytes sent by the remote terminal (not HKP).

`TLMPCNT` **(10.7.51)**

Attribute(s):

Brief: "1553 Telemetry Packet Count"

Instance(s): `tlmp`

A count of the number of Telem packets sent by the remote terminal (not HKP).

Used by:

`CpuMetr (10.4.3)`

10.5.8 `TMUX0` (TEM mux channel 0 environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 128 bits (16 bytes)

Description:

Describes 8 ADC quantities for TEM mux channel 0.

TEM mux channel 0 environmental ADCs (TMUX0) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	TKR25V: adc0															
002	TKR25I: adc1															
004	TKR15V: adc2															
006	TKR15I: adc3															
008	TKR25V: adc4															
00A	TKR25I: adc5															
00C	TKRBIASV: adc6															
00E	TKRBIASI: adc7															

```
TMUX0 (str)
|  TKR 2.5 Voltage ADC value adc0 (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  TKR 2.5 Current ADC value adc1 (same bf)
|  TKR 1.5 Voltage ADC value adc2 (same bf)
|  TKR 1.5 Current ADC value adc3 (same bf)
|  TKR 2.5 Voltage ADC value adc4 (same bf)
|  TKR 2.5 Current ADC value adc5 (same bf)
|  TKR Bias Voltage ADC value adc6 (same bf)
|  TKR Bias Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TMUX0 {

    /* TKR 2.5 Voltage ADC value */
    LHK_TKR25V           adc0;

    /* TKR 2.5 Current ADC value */
    LHK_TKR25I           adc1;

    /* TKR 1.5 Voltage ADC value */
    LHK_TKR15V           adc2;

    /* TKR 1.5 Current ADC value */
    LHK_TKR15I           adc3;

    /* TKR 2.5 Voltage ADC value */
    LHK_TKR25V           adc4;

    /* TKR 2.5 Current ADC value */
    LHK_TKR25I           adc5;

    /* TKR Bias Voltage ADC value */
    LHK_TKRBIASV         adc6;

    /* TKR Bias Current ADC value */
    LHK_TKRBIASI         adc7;

} LHK_TMUX0;
```

Fields:

`TKR15I` **(10.6.32)**

Attribute(s):

Brief: "TKR 1.5 Current ADC value"

Instance(s): `adc3`

Describes a TKR 1.5I ADC value and status.

`TKR15V` **(10.6.33)**

Attribute(s):

Brief: "TKR 1.5 Voltage ADC value"

Instance(s): `adc2`

Describes a TKR 1.5V ADC value and status.

`TKR25I` **(10.6.34)**

Attribute(s):

Brief: "TKR 2.5 Current ADC value"

Instance(s): `adc1`, `adc5`

Bitfield describing a TKR 2.5I ADC value and status

`TKR25V` **(10.6.35)**

Attribute(s):

Brief: "TKR 2.5 Voltage ADC value"

Instance(s): `adc0`, `adc4`

Bitfield describing a TRK 2.5V ADC value and status.

`TKRBIASI` **(10.6.36)**

Attribute(s):

Brief: "TKR Bias Current ADC value"

Instance(s): `adc7`

Describes a TKR bias current ADC value and status.

`TKRBIASV` **(10.6.37)**

Attribute(s):

Brief: "TKR Bias Voltage ADC value"

Instance(s): `adc6`

Describes a TKR bias voltage ADC value and status.

Used by:

`DiagTemEnvPwr0 (10.4.20)`

10.5.9 `TMUX1` (TEM mux channel 1 environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 128 bits (16 bytes)

Description:

Contains 6 CAL and 2 TEM power ADC environmental values.

TEM mux channel 1 environmental ADCs (TMUX1) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	CAL33V: adc0															
002	CAL33I: adc1															
004	CAL33V: adc2															
006	CAL33I: adc3															
008	CALBIASV: adc4															
00A	CALBIASI: adc5															
00C	TEM33V: adc6															
00E	TEM33I: adc7															

```
TMUX1 (str)
|  CAL 3.3 Voltage ADC value adc0 (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
|  CAL 3.3 Current ADC value adc1 (same bf)
|  CAL 3.3 Voltage ADC value adc2 (same bf)
|  CAL 3.3 Current ADC value adc3 (same bf)
|  CAL Bias Voltage ADC value adc4 (same bf)
|  CAL Bias Current ADC value adc5 (same bf)
|  TEM 3.3 Voltage ADC value adc6 (same bf)
|  TEM 3.3 Current ADC value adc7 (same bf)
```

```
typedef struct _LHK_TMUX1 {

    /* CAL 3.3 Voltage ADC value */
    LHK_CAL33V           adc0;

    /* CAL 3.3 Current ADC value */
    LHK_CAL33I           adc1;

    /* CAL 3.3 Voltage ADC value */
    LHK_CAL33V           adc2;

    /* CAL 3.3 Current ADC value */
    LHK_CAL33I           adc3;

    /* CAL Bias Voltage ADC value */
    LHK_CALBIASV         adc4;

    /* CAL Bias Current ADC value */
    LHK_CALBIASI         adc5;

    /* TEM 3.3 Voltage ADC value */
    LHK_TEM33V           adc6;

    /* TEM 3.3 Current ADC value */
    LHK_TEM33I           adc7;

} LHK_TMUX1;
```

Fields:

`CAL33I` **(10.6.12)**

Attribute(s):

Brief: "CAL 3.3 Current ADC value"

Instance(s): `adc1`, `adc3`

Describes a CAL 3.3I ADC value and status

`CAL33V` **(10.6.13)**

Attribute(s):

Brief: "CAL 3.3 Voltage ADC value"

Instance(s): `adc0`, `adc2`

Describes a CAL 3.3V ADC value and status

`CALBIASI` **(10.6.14)**

Attribute(s):

Brief: "CAL Bias Current ADC value"

Instance(s): `adc5`

Describes a CAL bias current ADC value and status

`CALBIASV` **(10.6.15)**

Attribute(s):

Brief: "CAL Bias Voltage ADC value"

Instance(s): `adc4`

Describes a CAL bias voltage ADC value and status

`TEM33I` **(10.6.27)**

Attribute(s):

Brief: "TEM 3.3 Current ADC value"

Instance(s): `adc7`

Describes a TEM 3.3 current ADC value and status.

`TEM33V` **(10.6.28)**

Attribute(s):

Brief: "TEM 3.3 Voltage ADC value"

Instance(s): `adc6`

Describes a TEM 3.3V ADC value and status.

Used by:

`DiagTemEnvPwr0 (10.4.20)`

10.5.10 `TMUX2` (TEM mux channel 2 environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes

Length: 128 bits (16 bytes)

Description:

Contains 8 AFFE ADC temperature values.

TEM mux channel 2 environmental ADCs (`TMUX2`) Telemetry Field Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	AFEETEMP: `adcs[0-7]`															

```
TMUX2 (str)
|  CAL AFFE temperature ADC value adcs[0-7] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
```

```
typedef struct _LHK_TMUX2 {

    /* CAL AFFE temperature ADC value */
    LHK_AFEETEMP         adcs[8];

} LHK_TMUX2;
```

Fields:

`AFEETEMP` **(10.6.11)**

Attribute(s):

Brief: "CAL AFFE temperature ADC value"

Instance(s): `adcs`

Describes a CAL AFFE temperature ADC value and status.

Used by:

`DiagTemEnvTemp0 (10.4.26)`

10.5.11 `TMUX3` (TEM mux channel 3 environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 128 bits (16 bytes)

Description:

Contains 8 TKR cable temperature ADC values.

TEM mux channel 3 environmental ADCs (`TMUX3`) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	TKRCBLT: `adcs[0-7]`															

```
TMUX3 (str)
|  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
```

```
typedef struct _LHK_TMUX3 {

    /* TKR Cable Temperature ADC value */
    LHK_TKRCBLT          adcs[8];

} LHK_TMUX3;
```

Fields:

`TKRCBLT` **(10.6.38)**

Attribute(s):

Brief: "TKR Cable Temperature ADC value"

Instance(s): `adcs`

Describes a TKR cable temperature ADC value and status.

Same as:

`TMUX4 (10.5.12)`

Used by:

`DiagTemEnvTemp0 (10.4.26)`

10.5.12 `TMUX4` (TEM mux channel 4 environmental ADCs) Telemetry Struct

Definition:

Alignment: 2 bytes
Length: 128 bits (16 bytes)

Description:

Contains 8 TKR cable temperature ADC value and status.

TEM mux channel 4 environmental ADCs (`TMUX4`) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	TKRCBLT: `adcs[0-7]`															

```
TMUX4 (str)
|  TKR Cable Temperature ADC value adcs[0-7] (bf)
|  |  ADC status bits stat (0-3)
|  |  ADC raw value val (4-15)
```

```
typedef struct _LHK_TMUX4 {

    /* TKR Cable Temperature ADC value */
    LHK_TKRCBLT          adcs[8];

} LHK_TMUX4;
```

Fields:

`TKRCBLT` **(10.6.38)**

Attribute(s):

Brief: "TKR Cable Temperature ADC value"

Instance(s): `adcs`

Describes a TKR cable temperature ADC value and status.

Same as:

`TMUX3 (10.5.11)`

Used by:

`DiagTemEnvTemp0 (10.4.26)`

10.6 Telemetry Bitfields

10.6.0 `ACDBEAGRIDTEMP` (ACD BEA Grid Interface Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
ACDBEAGRIDTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_ACDBEAGRIDTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_ACDBEAGRIDTEMP_Bfl;

typedef union _LHK_ACDBEAGRIDTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_ACDBEAGRIDTEMP_Bfl  bf;
```

```
} LHK_ACDBEAGRIDTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv3 (10.4.15)

10.6.1 ACDPMTRAILTEMP (ADC PMT Rail Temperature) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**3**	**4**	ADCS stat
4	**15**	**12**	ADCV val

```
ACDPMTRAILTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_ACDPMTRAILTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_ACDPMTRAILTEMP_Bfl;

typedef union _LHK_ACDPMTRAILTEMP {

    /* underlying unsigned short */
    unsigned short                   us;

    /* struct mapped bitfield */
    struct _LHK_ACDPMTRAILTEMP_Bfl  bf;

} LHK_ACDPMTRAILTEMP;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LAPMTRTEMPADCLIM (10.10.1)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0)`, `ACDSHELLTEMP (10.6.3)`, `AEMFR28ISUM (10.6.4)`, `AEMFR33ISUM (10.6.5)`, `AEMFRHV1 (10.6.6)`, `AEMFRHV2 (10.6.7)`, `AEMFRTEMP (10.6.9)`, `AEMFRVDD (10.6.10)`, `AFEETEMP (10.6.11)`, `CAL33I (10.6.12)`, `CAL33V (10.6.13)`, `CALBIASI (10.6.14)`, `CALBIASV (10.6.15)`, `CALBSPTEMP (10.6.16)`, `DABTEMP (10.6.17)`, `DAQ33V (10.6.18)`, `EPUTEMP (10.6.19)`, `EPUV (10.6.20)`, `GRIDRADIFTEMP (10.6.21)`, `GRIDTEMP (10.6.22)`, `PDUADC (???)`, `RADANHTRTEMP (10.6.25)`, `RADTEMP (10.6.26)`, `TEM33I (10.6.27)`, `TEM33V (10.6.28)`, `TEMPCBTEMP (10.6.30)`, `TEMPSTEMP (10.6.31)`, `TKR15I (10.6.32)`, `TKR15V (10.6.33)`, `TKR25I (10.6.34)`, `TKR25V (10.6.35)`, `TKRBIASI (10.6.36)`, `TKRBIASV (10.6.37)`, `TKRCBLT (10.6.38)`, `VCHPDSHPTEMP (10.6.39)`, `VCHPRSVRHTRTEMP (10.6.40)`, `VCHPXLHPTEMP (10.6.41)`

Used by:

`DiagPduEnv3 (10.4.15)`

10.6.2 `ACDPWRREG` (PDU ADC Power Management Register) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	12	13	`TSP13 spare`
13	13	1	`PDUACDCNVT cnvt`
14	14	1	`PDUACDPWRSUP sup`
15	15	1	`PDUACDPWRST acd`

```
ACDPWRREG (bf)
|  Spare 13 bit field spare (0-12)
|  PDU ACD Power Converter Source cnvt (13)
|  PDU ACD Power Supply Source sup (14)
|  PDU ACD Power State acd (15)
```

```
typedef struct _LHK_ACDPWRREG_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Spare 13 bit field */
    unsigned short       spare:13;

    /* PDU ACD Power Converter Source */
    unsigned short       cnvt:1;

    /* PDU ACD Power Supply Source */
    unsigned short       sup:1;

    /* PDU ACD Power State */
    unsigned short       acd:1;

    #elif ENDIANNESS_IS_LITTLE

    /* PDU ACD Power State */
    unsigned short       acd:1;

    /* PDU ACD Power Supply Source */
    unsigned short       sup:1;

    /* PDU ACD Power Converter Source */
    unsigned short       cnvt:1;

    /* Spare 13 bit field */
    unsigned short       spare:13;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_ACDPWRREG_Bfl;

typedef union _LHK_ACDPWRREG {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_ACDPWRREG_Bfl       bf;

} LHK_ACDPWRREG;
```

Fields:

PDUACDCNVT **(10.7.36)**

Attribute(s):
dsc: LAPDUPWRCNVTSTAT (10.9.1)

Brief: "PDU ACD Power Converter Source"

Instance(s): `cnvt`

`PDUACDPWRST` **(10.7.37)**

Attribute(s):
dsc: LAPDUPWRSTATES (10.9.2)

Brief: "PDU ACD Power State"

Instance(s): `acd`

`PDUACDPWRSUP` **(10.7.38)**

Attribute(s):
dsc: LAPDUPWRSUPSTAT (10.9.3)

Brief: "PDU ACD Power Supply Source"

Instance(s): `sup`

`TSP13` **(10.7.53)**

Attribute(s):

Brief: "Spare 13 bit field"

Instance(s): `spare`

Spare 13 bits

Used by:

`DiagPduEnv0 (10.4.12)`

10.6.3 `ACDSHELLTEMP` (ADC Shell Temperature) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

`ACDSHELLTEMP (bf)`

```
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_ACDSHELLTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_ACDSHELLTEMP_Bfl;

typedef union _LHK_ACDSHELLTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_ACDSHELLTEMP_Bfl    bf;

} LHK_ACDSHELLTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LASHLTEMPADCLIM (10.10.2)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)`

Used by:

`DiagPduEnv3 (10.4.15)`

10.6.4 `AEMFR28ISUM` (AEM FREE Board 28V Sum) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

The sum of the high voltage (28V) currents measured over all FREE boards.

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
AEMFR28ISUM (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_AEMFR28ISUM_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;
```

```
        /* ADC raw value */
        unsigned short       val:12;

        #elif ENDIANNESS_IS_LITTLE

        /* ADC raw value */
        unsigned short       val:12;

        /* ADC status bits */
        unsigned short       stat:4;

        #else

        #error _FILE_ ENDIANNESS NOT DEFINED

        #endif

    } __attribute__((packed)) LHK_AEMFR28ISUM_Bfl;

    typedef union _LHK_AEMFR28ISUM {

        /* underlying unsigned short */
        unsigned short                 us;

        /* struct mapped bitfield */
        struct _LHK_AEMFR28ISUM_Bfl     bf;

    } LHK_AEMFR28ISUM;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMDAQENV (10.5.0)

10.6.5 AEMFR33ISUM (AEM FREE Board 3.3 Sum) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

The sum of the digital(3.3) currents of all FREE boards.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
AEMFR33ISUM (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_AEMFR33ISUM_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else
```

```
    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_AEMFR33ISUM_Bfl;

typedef union _LHK_AEMFR33ISUM {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_AEMFR33ISUM_Bfl     bf;

} LHK_AEMFR33ISUM;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMDAQENV (10.5.0)

10.6.6 **AEMFRHV1** (AEM Free Board HV1) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
AEMFRHV1 (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_AEMFRHV1_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_AEMFRHV1_Bfl;

typedef union _LHK_AEMFRHV1 {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_AEMFRHV1_Bfl        bf;
```

```
} LHK_AEMFRHV1;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDAEMFRHV1ADCLIM (10.10.9)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMFRENV (10.5.1)

10.6.7 `AEMFRHV2` (AEM Free Board HV2) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
AEMFRHV2 (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_AEMFRHV2_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_AEMFRHV2_Bfl;

typedef union _LHK_AEMFRHV2 {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_AEMFRHV2_Bfl        bf;

} LHK_AEMFRHV2;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDAEMFRHV2ADCLIM (10.10.10)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMFRENV (10.5.1)

10.6.8 **AEMFRPWRREG** (AEM FREE board power status register) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Contains the status of the power supplied by the AEM to its FREE boards.

Start	Stop	Size	Item Type and Name
0	3	4	TSP4 spare
4	4	1	AEMFRPWRST free11
5	5	1	AEMFRPWRST free10
6	6	1	AEMFRPWRST free9
7	7	1	AEMFRPWRST free8
8	8	1	AEMFRPWRST free7
9	9	1	AEMFRPWRST free6

10	10	1	AEMFRPWRST free5
11	11	1	AEMFRPWRST free4
12	12	1	AEMFRPWRST free3
13	13	1	AEMFRPWRST free2
14	14	1	AEMFRPWRST free1
15	15	1	AEMFRPWRST free0

```
AEMFRPWRREG (bf)
|  Spare 4 bits spare (0-3)
|  AEM FREE Board Power State free11 (4) ... free0 (15)
```

```
typedef struct _LHK_AEMFRPWRREG_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Spare 4 bits */
    unsigned short       spare:4;

    /* AEM FREE Board Power State */
    unsigned short       free11:1;

    /* AEM FREE Board Power State */
    unsigned short       free10:1;

    /* AEM FREE Board Power State */
    unsigned short       free9:1;

    /* AEM FREE Board Power State */
    unsigned short       free8:1;

    /* AEM FREE Board Power State */
    unsigned short       free7:1;

    /* AEM FREE Board Power State */
    unsigned short       free6:1;

    /* AEM FREE Board Power State */
    unsigned short       free5:1;

    /* AEM FREE Board Power State */
    unsigned short       free4:1;

    /* AEM FREE Board Power State */
    unsigned short       free3:1;

    /* AEM FREE Board Power State */
    unsigned short       free2:1;

    /* AEM FREE Board Power State */
    unsigned short       free1:1;

    /* AEM FREE Board Power State */
    unsigned short       free0:1;
```

```
    #elif ENDIANNESS_IS_LITTLE

    /* AEM FREE Board Power State */
    unsigned short       free0:1;

    /* AEM FREE Board Power State */
    unsigned short       free1:1;

    /* AEM FREE Board Power State */
    unsigned short       free2:1;

    /* AEM FREE Board Power State */
    unsigned short       free3:1;

    /* AEM FREE Board Power State */
    unsigned short       free4:1;

    /* AEM FREE Board Power State */
    unsigned short       free5:1;

    /* AEM FREE Board Power State */
    unsigned short       free6:1;

    /* AEM FREE Board Power State */
    unsigned short       free7:1;

    /* AEM FREE Board Power State */
    unsigned short       free8:1;

    /* AEM FREE Board Power State */
    unsigned short       free9:1;

    /* AEM FREE Board Power State */
    unsigned short       free10:1;

    /* AEM FREE Board Power State */
    unsigned short       free11:1;

    /* Spare 4 bits */
    unsigned short       spare:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_AEMFRPWRREG_Bfl;

typedef union _LHK_AEMFRPWRREG {

    /* underlying unsigned short */
    unsigned short                 us;
```

```
    /* struct mapped bitfield */
    struct _LHK_AEMFRPWRREG_Bfl     bf;

} LHK_AEMFRPWRREG;
```

Fields:

`AEMFRPWRST` **(10.7.2)**

Attribute(s):
dsc: LAEMFRPWRSTATES (10.9.0)

Brief: "AEM FREE Board Power State"

Instance(s): `free0`, `free1`, `free10`, `free11`, `free2`, `free3`, `free4`, `free5`, `free6`, `free7`, `free8`, `free9`

Describes the AEM free board power status.

`TSP4` **(10.7.55)**

Attribute(s):

Brief: "Spare 4 bits"

Instance(s): `spare`

Spare 4 bits

Used by:

`AemEnv0 (10.4.0)`

10.6.9 **`AEMFRTEMP`** (AEM Free board Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
AEMFRTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_AEMFRTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_AEMFRTEMP_Bfl;

typedef union _LHK_AEMFRTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_AEMFRTEMP_Bfl       bf;

} LHK_AEMFRTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDAEMFRTMPADCLIM (10.10.11)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMFRENV (10.5.1)

10.6.10 AEMFRVDD (AEM VDD) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
AEMFRVDD (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_AEMFRVDD_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE
```

```
        /* ADC raw value */
        unsigned short       val:12;

        /* ADC status bits */
        unsigned short       stat:4;

        #else

        #error _FILE_ ENDIANNESS NOT DEFINED

        #endif

    } __attribute__((packed)) LHK_AEMFRVDD_Bfl;

    typedef union _LHK_AEMFRVDD {

        /* underlying unsigned short */
        unsigned short                  us;

        /* struct mapped bitfield */
        struct _LHK_AEMFRVDD_Bfl        bf;

    } LHK_AEMFRVDD;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDAEMFRVDDADCLIM (10.10.12)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2

(10.6.7), AEMFRTEMP (10.6.9), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMFRENV (10.5.1)

10.6.11 AFEETEMP (CAL AFFE temperature ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a CAL AFFE temperature ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
AFEETEMP (bf)
| ADC status bits stat (0-3)
| ADC raw value val (4-15)

typedef struct _LHK_AFEETEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif
```

```
} __attribute__((packed)) LHK_AFEETEMP_Bfl;

typedef union _LHK_AFEETEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_AFEETEMP_Bfl        bf;

} LHK_AFEETEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LCAFETADCLIM (10.10.5)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX2 (10.5.10)

10.6.12 `CAL33I` (CAL 3.3 Current ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a CAL 3.3I ADC value and status

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
CAL33I (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_CAL33I_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_CAL33I_Bfl;

typedef union _LHK_CAL33I {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_CAL33I_Bfl          bf;

} LHK_CAL33I;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LC33IADCLIM (10.10.3)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX1 (10.5.9)

10.6.13 `CAL33V` (CAL 3.3 Voltage ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a CAL 3.3V ADC value and status

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
CAL33V (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_CAL33V_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_CAL33V_Bfl;

typedef union _LHK_CAL33V {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_CAL33V_Bfl         bf;

} LHK_CAL33V;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LC33VADCLIM (10.10.4)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX1 (10.5.9)

10.6.14 `CALBIASI` (CAL Bias Current ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a CAL bias current ADC value and status

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
CALBIASI (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_CALBIASI_Bfl {
```

```
    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_CALBIASI_Bfl;

typedef union _LHK_CALBIASI {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_CALBIASI_Bfl       bf;

} LHK_CALBIASI;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LCBIASIADCLIM (10.10.7)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX1 (10.5.9)

10.6.15 CALBIASV (CAL Bias Voltage ADC value) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Describes a CAL bias voltage ADC value and status

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
CALBIASV (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_CALBIASV_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
```

```
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_CALBIASV_Bfl;

typedef union _LHK_CALBIASV {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_CALBIASV_Bfl       bf;

} LHK_CALBIASV;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LCBIASVADCLIM (10.10.8)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I`

(10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX1 (10.5.9)

10.6.16 **CALBSPTEMP** (CAL Baseplate Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
CALBSPTEMP (bf)
|  ADC status bits stat (0–3)
|  ADC raw value val (4–15)
```

```
typedef struct _LHK_CALBSPTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_CALBSPTEMP_Bfl;
```

```
typedef union _LHK_CALBSPTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_CALBSPTEMP_Bfl      bf;

} LHK_CALBSPTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LCBASPLADCLIM (10.10.6)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv1 (10.4.13)

10.6.17 **DABTEMP** (DAQ Board Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

The temperature of the DAQ board.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
DABTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_DABTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_DABTEMP_Bfl;

typedef union _LHK_DABTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_DABTEMP_Bfl         bf;

} LHK_DABTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMDAQENV (10.5.0)

10.6.18 `DAQ33V` (DAQ Board 3.3 Voltage) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

The digital (3.3) voltage of the DAQ board.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat

4 15 12 `ADCV val`

```
DAQ33V (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_DAQ33V_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_DAQ33V_Bfl;

typedef union _LHK_DAQ33V {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_DAQ33V_Bfl         bf;

} LHK_DAQ33V;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

AEMDAQENV (10.5.0)

10.6.19 `EPUTEMP` (EPU Temperature ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
EPUTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_EPUTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;
```

```
    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_EPUTEMP_Bfl;

typedef union _LHK_EPUTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_EPUTEMP_Bfl         bf;

} LHK_EPUTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDEPUTEMPADCLIM (10.10.13)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv0 (10.4.12)

10.6.20 **EPUV** (EPU Voltage ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
EPUV (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_EPUV_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else
```

```
    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_EPUV_Bfl;

typedef union _LHK_EPUV {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_EPUV_Bfl            bf;

} LHK_EPUV;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDEPUVADCLIM (10.10.14)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv0 (10.4.12)

10.6.21 GRIDRADIFTEMP (Grid Radiator Interface Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
GRIDRADIFTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_GRIDRADIFTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_GRIDRADIFTEMP_Bfl;

typedef union _LHK_GRIDRADIFTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
```

```
    struct _LHK_GRIDRADIFTEMP_Bfl   bf;

} LHK_GRIDRADIFTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMGRDRADIFADCLIM (10.10.19)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCH-PRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv3 (10.4.15)

10.6.22 `GRIDTEMP` (Grid Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
GRIDTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_GRIDTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_GRIDTEMP_Bfl;

typedef union _LHK_GRIDTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_GRIDTEMP_Bfl        bf;

} LHK_GRIDTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMGRIDTEMPADCLIM (10.10.20)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)`

Used by:

`DiagPduEnv3 (10.4.15)`

10.6.23 `PDUEPUCRATEPWR` (PDU EPU Crate Power States) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	9	10	`TSP10 spare0`
10	10	1	`PDUEPUCVTST epu2cvt`
11	11	1	`PDUEPUCVTST epu1cvt`
12	12	1	`PDUEPUCVTST epu0cvt`
13	13	1	`PDUEPUPWRST epu2pwr`
14	14	1	`PDUEPUPWRST epu1pwr`
15	15	1	`PDUEPUPWRST epu0pwr`

`PDUEPUCRATEPWR (bf)`

```
| Spare 10 bits spare0 (0-9)
| PDU EPU Converter State epu2cvt (10) ... epu0cvt (12)
| PDU EPU Crate Power State epu2pwr (13) ... epu0pwr (15)

typedef struct _LHK_PDUEPUCRATEPWR_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Spare 10 bits */
    unsigned short       spare0:10;

    /* PDU EPU Converter State */
    unsigned short       epu2cvt:1;

    /* PDU EPU Converter State */
    unsigned short       epu1cvt:1;

    /* PDU EPU Converter State */
    unsigned short       epu0cvt:1;

    /* PDU EPU Crate Power State */
    unsigned short       epu2pwr:1;

    /* PDU EPU Crate Power State */
    unsigned short       epu1pwr:1;

    /* PDU EPU Crate Power State */
    unsigned short       epu0pwr:1;

    #elif ENDIANNESS_IS_LITTLE

    /* PDU EPU Crate Power State */
    unsigned short       epu0pwr:1;

    /* PDU EPU Crate Power State */
    unsigned short       epu1pwr:1;

    /* PDU EPU Crate Power State */
    unsigned short       epu2pwr:1;

    /* PDU EPU Converter State */
    unsigned short       epu0cvt:1;

    /* PDU EPU Converter State */
    unsigned short       epu1cvt:1;

    /* PDU EPU Converter State */
    unsigned short       epu2cvt:1;

    /* Spare 10 bits */
    unsigned short       spare0:10;

    #else
```

```
    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_PDUEPUCRATEPWR_Bfl;

typedef union _LHK_PDUEPUCRATEPWR {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_PDUEPUCRATEPWR_Bfl  bf;

} LHK_PDUEPUCRATEPWR;
```

Fields:

`PDUEPUCVTST` **(10.7.39)**

Attribute(s):
dsc: LDPDUEPUCNVT (10.9.4)

Brief: "PDU EPU Converter State"

Instance(s): `epu0cvt`, `epu1cvt`, `epu2cvt`

`PDUEPUPWRST` **(10.7.40)**

Attribute(s):
dsc: LDPDUEPUPWRST (10.9.5)

Brief: "PDU EPU Crate Power State"

Instance(s): `epu0pwr`, `epu1pwr`, `epu2pwr`

`TSP10` **(10.7.52)**

Attribute(s):

Brief: "Spare 10 bits"

Instance(s): `spare0`

Used by:

`DiagPduEnv0 (10.4.12)`

10.6.24 PDUTEMPWRREG (PDU TEM Power Management Register) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	0	1	PDUTEMPWRST temf
1	1	1	PDUTEMPWRST teme
2	2	1	PDUTEMPWRST temd
3	3	1	PDUTEMPWRST temc
4	4	1	PDUTEMPWRST temb
5	5	1	PDUTEMPWRST tema
6	6	1	PDUTEMPWRST tem9
7	7	1	PDUTEMPWRST tem8
8	8	1	PDUTEMPWRST tem7
9	9	1	PDUTEMPWRST tem6
10	10	1	PDUTEMPWRST tem5
11	11	1	PDUTEMPWRST tem4
12	12	1	PDUTEMPWRST tem3
13	13	1	PDUTEMPWRST tem2
14	14	1	PDUTEMPWRST tem1
15	15	1	PDUTEMPWRST tem0

```
PDUTEMPWRREG (bf)
|  PDU TEM Power State temf (0) ... tem0 (15)

typedef struct _LHK_PDUTEMPWRREG_Bfl {

    #if ENDIANNESS_IS_BIG

    /* PDU TEM Power State */
    unsigned short       temf:1;

    /* PDU TEM Power State */
    unsigned short       teme:1;

    /* PDU TEM Power State */
    unsigned short       temd:1;

    /* PDU TEM Power State */
    unsigned short       temc:1;

    /* PDU TEM Power State */
    unsigned short       temb:1;

    /* PDU TEM Power State */
    unsigned short       tema:1;

    /* PDU TEM Power State */
    unsigned short       tem9:1;
```

```
/* PDU TEM Power State */
unsigned short       tem8:1;

/* PDU TEM Power State */
unsigned short       tem7:1;

/* PDU TEM Power State */
unsigned short       tem6:1;

/* PDU TEM Power State */
unsigned short       tem5:1;

/* PDU TEM Power State */
unsigned short       tem4:1;

/* PDU TEM Power State */
unsigned short       tem3:1;

/* PDU TEM Power State */
unsigned short       tem2:1;

/* PDU TEM Power State */
unsigned short       tem1:1;

/* PDU TEM Power State */
unsigned short       tem0:1;

#elif ENDIANNESS_IS_LITTLE

/* PDU TEM Power State */
unsigned short       tem0:1;

/* PDU TEM Power State */
unsigned short       tem1:1;

/* PDU TEM Power State */
unsigned short       tem2:1;

/* PDU TEM Power State */
unsigned short       tem3:1;

/* PDU TEM Power State */
unsigned short       tem4:1;

/* PDU TEM Power State */
unsigned short       tem5:1;

/* PDU TEM Power State */
unsigned short       tem6:1;

/* PDU TEM Power State */
unsigned short       tem7:1;

/* PDU TEM Power State */
```

```
    unsigned short       tem8:1;

    /* PDU TEM Power State */
    unsigned short       tem9:1;

    /* PDU TEM Power State */
    unsigned short       tema:1;

    /* PDU TEM Power State */
    unsigned short       temb:1;

    /* PDU TEM Power State */
    unsigned short       temc:1;

    /* PDU TEM Power State */
    unsigned short       temd:1;

    /* PDU TEM Power State */
    unsigned short       teme:1;

    /* PDU TEM Power State */
    unsigned short       temf:1;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_PDUTEMPWRREG_Bfl;

typedef union _LHK_PDUTEMPWRREG {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_PDUTEMPWRREG_Bfl    bf;

} LHK_PDUTEMPWRREG;
```

Fields:

`PDUTEMPWRST` **(10.7.41)**

Attribute(s):
dsc: LDPDUTEMPWRST (10.9.6)

Brief: "PDU TEM Power State"

Instance(s): `tem0`, `tem1`, `tem2`, `tem3`, `tem4`, `tem5`, `tem6`, `tem7`, `tem8`, `tem9`, `tema`, `temb`, `temc`, `temd`, `teme`, `temf`

Used by:

DiagPduEnv0 (10.4.12)

10.6.25 RADANHTRTEMP (Radiator Anitfreeze Heater Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
RADANHTRTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_RADANHTRTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_RADANHTRTEMP_Bfl;

typedef union _LHK_RADANHTRTEMP {

    /* underlying unsigned short */
    unsigned short                     us;

    /* struct mapped bitfield */
```

```
    struct _LHK_RADANHTRTEMP_Bfl    bf;

} LHK_RADANHTRTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMRADAFHTRADCLIM (10.10.21)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv3 (10.4.15)

10.6.26 `RADTEMP` (Radiator Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
RADTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_RADTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_RADTEMP_Bfl;

typedef union _LHK_RADTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_RADTEMP_Bfl         bf;

} LHK_RADTEMP;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMRADTEMPADCLIM (10.10.22)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)`

Used by:

`DiagPduEnv3 (10.4.15)`

10.6.27 `TEM33I` (TEM 3.3 Current ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a TEM 3.3 current ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
TEM33I (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TEM33I_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TEM33I_Bfl;

typedef union _LHK_TEM33I {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_TEM33I_Bfl          bf;

} LHK_TEM33I;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDTEM33IADCLIM (10.10.15)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX1 (10.5.9)

10.6.28 `TEM33V` (TEM 3.3 Voltage ADC value) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Describes a TEM 3.3V ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TEM33V (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TEM33V_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;
```

```
    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TEM33V_Bfl;

typedef union _LHK_TEM33V {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_TEM33V_Bfl         bf;

} LHK_TEM33V;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
alg: LDTEMVADCCNV (10.8.0)
lim: LDTEM33VADCLIM (10.10.16)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv0 (10.4.12)

10.6.29 TEMDEADTIMEREG (TEM Deadtime Register Bitfield) Telemetry Bitfield

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Describes the TEM dead time low-rate science counter register.

Start	Stop	Size	Item Type and Name
0	23	24	TEMDEADTIME deadtime

```
TEMDEADTIMEREG (bf)
|  TEM Deadtime Low-rate Science Counter deadtime (0-23)
```

```
typedef struct _LHK_TEMDEADTIMEREG_Bfl {

    #if ENDIANNESS_IS_BIG

    /* TEM Deadtime Low-rate Science Counter */
    unsigned int         deadtime:24;

    /* Bit pad */
    unsigned int         bit_pad:8;

    #elif ENDIANNESS_IS_LITTLE

    /* Bit pad */
    unsigned int         bit_pad:8;

    /* TEM Deadtime Low-rate Science Counter */
    unsigned int         deadtime:24;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED
```

```
    #endif

} __attribute__((packed)) LHK_TEMDEADTIMEREG_Bfl;

typedef union _LHK_TEMDEADTIMEREG {

    /* underlying unsigned int */
    unsigned int                    ui;

    /* struct mapped bitfield */
    struct _LHK_TEMDEADTIMEREG_Bfl  bf;

} LHK_TEMDEADTIMEREG;
```

Fields:

`TEMDEADTIME` **(10.7.47)**

Attribute(s):

Brief: "TEM Deadtime Low-rate Science Counter"

Instance(s): `deadtime`

This register counts the deadtime incurred by specific trigger primitives.

Used by:

`DiagLrs0 (10.4.9)`

10.6.30 `TEMPCBTEMP` (TEM PCB Temperature) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**3**	**4**	`ADCS stat`
4	**15**	**12**	`ADCV val`

```
TEMPCBTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_TEMPCBTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
```

```
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TEMPCBTEMP_Bfl;

typedef union _LHK_TEMPCBTEMP {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_TEMPCBTEMP_Bfl      bf;

} LHK_TEMPCBTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDTEMPCBTADCLIM (10.10.17)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv1 (10.4.13)

10.6.31 **TEMPSTEMP** (TEM Power Supply Temperature ADC value) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TEMPSTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_TEMPSTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;
```

```
    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TEMPSTEMP_Bfl;

typedef union _LHK_TEMPSTEMP {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_TEMPSTEMP_Bfl       bf;

} LHK_TEMPSTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LDTEMPSTADCLIM (10.10.18)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20),

GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv0 (10.4.12)

10.6.32 TKR15I (TKR 1.5 Current ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a TKR 1.5I ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TKR15I (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKR15I_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKR15I_Bfl;
```

```
typedef union _LHK_TKR15I {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_TKR15I_Bfl          bf;

} LHK_TKR15I;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LT15IADCLIM (10.10.26)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSH-PTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.33 **TKR15V** (TKR 1.5 Voltage ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a TKR 1.5V ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TKR15V (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKR15V_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKR15V_Bfl;

typedef union _LHK_TKR15V {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_TKR15V_Bfl         bf;

} LHK_TKR15V;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): stat

A 4 bit value representing an ADC read/evaluation status.

ADCV **(10.7.1)**

Attribute(s):
lim: LT15VADCLIM (10.10.27)

Brief: "ADC raw value"

Instance(s): val

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSH-PTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.34 TKR25I (TKR 2.5 Current ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield describing a TKR 2.5I ADC value and status

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TKR25I (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKR25I_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKR25I_Bfl;

typedef union _LHK_TKR25I {

    /* underlying unsigned short */
    unsigned short                   us;

    /* struct mapped bitfield */
    struct _LHK_TKR25I_Bfl           bf;

} LHK_TKR25I;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LT25IADCLIM (10.10.28)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSH-PTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.35 `TKR25V` (TKR 2.5 Voltage ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield describing a TRK 2.5V ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
TKR25V (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKR25V_Bfl {
```

```
    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKR25V_Bfl;

typedef union _LHK_TKR25V {

    /* underlying unsigned short */
    unsigned short                 us;

    /* struct mapped bitfield */
    struct _LHK_TKR25V_Bfl         bf;

} LHK_TKR25V;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): stat

A 4 bit value representing an ADC read/evaluation status.

ADCV **(10.7.1)**

Attribute(s):
lim: LT25VADCLIM (10.10.29)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSH-PTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.36 `TKRBIASI` (TKR Bias Current ADC value) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Describes a TKR bias current ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TKRBIASI (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKRBIASI_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
```

```
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKRBIASI_Bfl;

typedef union _LHK_TKRBIASI {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_TKRBIASI_Bfl        bf;

} LHK_TKRBIASI;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LTBIASIADCLIM (10.10.30)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I`

(10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.37 TKRBIASV (TKR Bias Voltage ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a TKR bias voltage ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
TKRBIASV (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_TKRBIASV_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif
```

```
} __attribute__((packed)) LHK_TKRBIASV_Bfl;

typedef union _LHK_TKRBIASV {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_TKRBIASV_Bfl        bf;

} LHK_TKRBIASV;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LTBIASVADCLIM (10.10.31)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX0 (10.5.8)

10.6.38 `TKRCBLT` (TKR Cable Temperature ADC value) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Describes a TKR cable temperature ADC value and status.

Start	Stop	Size	Item Type and Name
0	3	4	`ADCS stat`
4	15	12	`ADCV val`

```
TKRCBLT (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_TKRCBLT_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_TKRCBLT_Bfl;

typedef union _LHK_TKRCBLT {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_TKRCBLT_Bfl         bf;

} LHK_TKRCBLT;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LTCBLTADCLIM (10.10.32)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)

Used by:

TMUX3 (10.5.11), TMUX4 (10.5.12)

10.6.39 `VCHPDSHPTEMP` (VCHP-DSHP Interface Temperature) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name

0	**3**	**4**	ADCS stat
4	**15**	**12**	ADCV val

```
VCHPDSHPTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)

typedef struct _LHK_VCHPDSHPTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_VCHPDSHPTEMP_Bfl;

typedef union _LHK_VCHPDSHPTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_VCHPDSHPTEMP_Bfl    bf;

} LHK_VCHPDSHPTEMP;
```

Fields:

ADCS **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMVCHPDSHPADCLIM (10.10.23)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

`ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41)`

Used by:

`DiagPduEnv2 (10.4.14)`

10.6.40 `VCHPRSVRHTRTEMP` (VCHP Reservoir Heater Temperature) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	**3**	**4**	`ADCS stat`
4	**15**	**12**	`ADCV val`

```
VCHPRSVRHTRTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_VCHPRSVRHTRTEMP_Bfl {

    #if ENDIANNESS_IS_BIG
```

```
    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
    unsigned short       stat:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LHK_VCHPRSVRHTRTEMP_Bfl;

typedef union _LHK_VCHPRSVRHTRTEMP {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LHK_VCHPRSVRHTRTEMP_Bfl bf;

} LHK_VCHPRSVRHTRTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMVCHPRSVTADCLIM (10.10.24)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38), VCHPDSHPTEMP (10.6.39), VCHPXLHPTEMP (10.6.41)

Used by:

DiagPduEnv2 (10.4.14)

10.6.41 VCHPXLHPTEMP (VCHP-XLHP Interface Temperature) Telemetry Bitfield

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	ADCS stat
4	15	12	ADCV val

```
VCHPXLHPTEMP (bf)
|  ADC status bits stat (0-3)
|  ADC raw value val (4-15)
```

```
typedef struct _LHK_VCHPXLHPTEMP_Bfl {

    #if ENDIANNESS_IS_BIG

    /* ADC status bits */
    unsigned short       stat:4;

    /* ADC raw value */
    unsigned short       val:12;

    #elif ENDIANNESS_IS_LITTLE

    /* ADC raw value */
    unsigned short       val:12;

    /* ADC status bits */
```

```
        unsigned short       stat:4;

        #else

        #error _FILE_ ENDIANNESS NOT DEFINED

        #endif

    } __attribute__((packed)) LHK_VCHPXLHPTEMP_Bfl;

    typedef union _LHK_VCHPXLHPTEMP {

        /* underlying unsigned short */
        unsigned short                 us;

        /* struct mapped bitfield */
        struct _LHK_VCHPXLHPTEMP_Bfl    bf;

    } LHK_VCHPXLHPTEMP;
```

Fields:

`ADCS` **(10.7.0)**

Attribute(s):
dsc: LHKSTATUSBITS (10.9.7)

Brief: "ADC status bits"

Instance(s): `stat`

A 4 bit value representing an ADC read/evaluation status.

`ADCV` **(10.7.1)**

Attribute(s):
lim: LMVCHPXLHPADCLIM (10.10.25)

Brief: "ADC raw value"

Instance(s): `val`

A 12 bit raw ADC value.

Same as:

ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), AEMFR28ISUM (10.6.4), AEMFR33ISUM (10.6.5), AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), AFEETEMP (10.6.11), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), CALBSPTEMP (10.6.16), DABTEMP (10.6.17), DAQ33V (10.6.18), EPUTEMP (10.6.19), EPUV (10.6.20), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), PDUADC (???), RADANHTRTEMP

(10.6.25), RADTEMP (10.6.26), TEM33I (10.6.27), TEM33V (10.6.28), TEMPCBTEMP (10.6.30), TEMPSTEMP (10.6.31), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKRBIASV (10.6.37), TKRCBLT (10.6.38),VCHPDSHPTEMP (10.6.39),VCHPRSVRHTRTEMP (10.6.40)

Used by:

DiagPduEnv2 (10.4.14)

10.7 Telemetry Fields

10.7.0 `ADCS` (ADC status bits) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	4 bits

Description:

A 4 bit value representing an ADC read/evaluation status.

Used by:

`AEMFRHV1 (10.6.6),AEMFRHV2 (10.6.7),AEMFRTEMP (10.6.9),AEMFRVDD (10.6.10)`

10.7.1 `ADCV` (ADC raw value) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	12 bits

Description:

A 12 bit raw ADC value.

Used by:

`AEMFRHV1 (10.6.6),AEMFRHV2 (10.6.7),AEMFRTEMP (10.6.9),AEMFRVDD (10.6.10)`

10.7.2 `AEMFRPWRST` (AEM FREE Board Power State) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	1 bit

Description:

Describes the AEM free board power status.

Used by:

`AEMFRPWRREG (10.6.8)`

10.7.3 `CMDCNTS` (Commands sent counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Commands sent to task.

Used by:

`CMDCNTRS (10.5.2)`

10.7.4 `CMDDISPF` (Command dispatch failure counter) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Commands failing task dispatch.

Used by:

`CMDCNTRS (10.5.2)`

10.7.5 `CMDEXEF` (Command execution failure counter) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Commands failing task execution

Used by:

`CMDCNTRS (10.5.2)`

10.7.6 `CPUJT` (CPU Junction Temperature) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

The CPU junction temperature, in degress Celcius.

Used by:

`CPUMETR (10.5.3)`

10.7.7 `CRXBCNT` (1553 Cmd Rx Byte Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of CmdRx bytes received by the remote terminal

Used by:

`RTSTATS (10.5.7)`

10.7.8 `CRXPCNT` (1553 Cmd Rx Packet Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of CmdRx packets received by the remote terminal

Used by:

`RTSTATS (10.5.7)`

10.7.9 `CTXBCNT` (1553 Cmd Tx Byte Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of CmdTx bytes sent by the remote terminal.

Used by:

`RTSTATS (10.5.7)`

10.7.10 `CTXPCNT` (1553 Cmd Tx Packet Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of CmdTx packets sent by the remote terminal

Used by:

`RTSTATS (10.5.7)`

10.7.11 `FILEIDCOM` (FILE ID Commit) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The file ID value for commiting the file data to storage.

Used by:

`FILESTATS (10.5.4)`

10.7.12 `FILEPKTCNT` (FILE Packet Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The number of vaild data packets received for the current file upload.

Used by:

`FILESTATS (10.5.4)`

10.7.13 `FILERRCNT` (FILE Error Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of errors reported for the current file upload.

Used by:

`FILESTATS (10.5.4)`

10.7.14 `FILERRCODE` (FILE Error Code) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The current file upload error indicator.

Used by:

`FILESTATS (10.5.4)`

10.7.15 `FILESIZECUR` (File Size Current) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The number of data bytes received for the current file upload.

Used by:

`FILESTATS (10.5.4)`

10.7.16 `FILESTATE` (FILE Upload State) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

The current file upload state.

Used by:

`FILESTATS (10.5.4)`

10.7.17 `GEMLRSDISC` (GEM low-rate science discarded counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Window turns discarded due to busy

Used by:

`GEMLRS (10.5.5)`

10.7.18 `GEMLRSLIVE` (GEM low-rate science livetime counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

1/deadtime statistics

Used by:

`GEMLRS (10.5.5)`

10.7.19 `GEMLRSSENT` (GEM low-rate science sent counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Window turns which result in a sent message

Used by:

`GEMLRS (10.5.5)`

10.7.20 `GEMPRSCL` (Low-rate science prescaled counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Window turns discarded due to prescaling

Used by:

`GEMLRS (10.5.5)`

10.7.21 `HKBCNT` (1553 Housekeeping Byte Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of HKP bytes sent by the remote terminal

Used by:

`RTSTATS (10.5.7)`

10.7.22 `HKPCNT` (1553 Housekeeping Packet Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of HKP packets sent by the remote terminal

Used by:

`RTSTATS (10.5.7)`

10.7.23 `INTRCNT` (1553 Interrupts) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of remote terminal device interrupts.

Used by:

`RTSTATS (10.5.7)`

10.7.24 `MDPACT` (Memory dump active) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Nonzero if dump is active.

Used by:

`MEMSTATS (10.5.6)`

10.7.25 `MDPADDR` (Memory dump address) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Current dump address.

Used by:

`MEMSTATS (10.5.6)`

10.7.26 `MDPBYTS` (Memory dump bytes) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Total number of bytes to dump.

Used by:

`MEMSTATS (10.5.6)`

10.7.27 `MDPFCD` (Memory dump function code) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Function code for current dump.

Used by:

`MEMSTATS (10.5.6)`

10.7.28 `MDPSTADR` (Memory dump start address) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Starting dump address.

Used by:

`MEMSTATS (10.5.6)`

10.7.29 `MDPSTAT` (Memory dump status) Telemetry Field

Definition:

Alignment: 4 bytes
C type: unsigned int
Length: 32 bits (4 bytes)

Description:

Status of most recent dump action.

Used by:

`MEMSTATS (10.5.6)`

10.7.30 `MDPTXID` (Memory dump transaction ID) Telemetry Field

Definition:

Alignment: 4 bytes
C type: unsigned int
Length: 32 bits (4 bytes)

Description:

Transaction ID for current dump.

Used by:

`MEMSTATS (10.5.6)`

10.7.31 `MLDACT` (Memory load active flag) Telemetry Field

Definition:

Alignment: 4 bytes
C type: unsigned int
Length: 32 bits (4 bytes)

Description:

Nonzero if load is active

Used by:

`MEMSTATS (10.5.6)`

10.7.32 `MLDBYTS` (Memory load total bytes) Telemetry Field

Definition:

Alignment: 4 bytes
C type: unsigned int
Length: 32 bits (4 bytes)

Description:

Total number of bytes to load.

Used by:

`MEMSTATS (10.5.6)`

10.7.33 **`MLDOFF`** (Memory load offset) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Current load offset.

Used by:

`MEMSTATS (10.5.6)`

10.7.34 **`MLDSTADR`** (Starting memory load address) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`MEMSTATS (10.5.6)`

10.7.35 **`MLDSTAT`** (Status of most recent load action) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`MEMSTATS (10.5.6)`

10.7.36 **`PDUACDCNVT`** (PDU ACD Power Converter Source) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	1 bit

Description:

Used by:

ACDPWRREG (10.6.2)

10.7.37 **PDUACDPWRST** (PDU ACD Power State) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Used by:

ACDPWRREG (10.6.2)

10.7.38 **PDUACDPWRSUP** (PDU ACD Power Supply Source) Telemetry Field

Definition:

Alignment:	1 byte
C type:	char
Length:	1 bit

Description:

Used by:

ACDPWRREG (10.6.2)

10.7.39 **PDUEPUCVTST** (PDU EPU Converter State) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	1 bit

Description:

Used by:

PDUEPUCRATEPWR (10.6.23)

10.7.40 **PDUEPUPWRST** (PDU EPU Crate Power State) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	1 bit

Description:

Used by:

`PDUEPUCRATEPWR (10.6.23)`

10.7.41 PDUTEMPWRST (PDU TEM Power State) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	1 bit

Description:

Used by:

`PDUTEMPWRREG (10.6.24)`

10.7.42 RLADC (Red Limit ADC Number) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

ACD number that exceeded its red limit.

Used by:

`RedLimAlrt (10.4.46)`

10.7.43 RLCNT (Red Limit ADC Counts) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

This field is used to report the raw ADC count value that exceeded the specified red limit.

Used by:

`RedLimAlrt (10.4.46)`

10.7.44 RLDEV (Red Limit Device Opcode) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short

Length: 16 bits (2 bytes)

Description:

Opcode specifying the device that exceeded a red limit.

Used by:

`RedLimAlrt (10.4.46)`

10.7.45 `RLLIM` (Red Limit Threshold) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

This field reports the limit threshold that was used to evaluate the exceeded ADC value.

Used by:

`RedLimAlrt (10.4.46)`

10.7.46 `RTERR` (1553 Error Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of remote terminal errors.

Used by:

`RTSTATS (10.5.7)`

10.7.47 `TEMDEADTIME` (TEM Deadtime Low-rate Science Counter) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	24 bits

Description:

This register counts the deadtime incurred by specific trigger primitives.

Used by:

`TEMDEADTIMEREG (10.6.29)`

10.7.48 `TIMESEC` (Timestamp seconds) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`CMDCNTRS (10.5.2)`

10.7.49 `TIMESUBSEC` (Timestamp subseconds) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`CMDCNTRS (10.5.2)`

10.7.50 `TLMBCNT` (1553 Telemertry Byte Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of Telem bytes sent by the remote terminal (not HKP).

Used by:

`RTSTATS (10.5.7)`

10.7.51 `TLMPCNT` (1553 Telemetry Packet Count) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

A count of the number of Telem packets sent by the remote terminal (not HKP).

Used by:

`RTSTATS (10.5.7)`

10.7.52 `TSP10` (Spare 10 bits) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	10 bits

Description:

Used by:

`PDUEPUCRATEPWR (10.6.23)`

10.7.53 `TSP13` (Spare 13 bit field) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	13 bits

Description:

Spare 13 bits

Used by:

`ACDPWRREG (10.6.2)`

10.7.54 `TSP16` (Spare 16 bit field) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`CmdCnt0 (10.4.1)`

10.7.55 `TSP4` (Spare 4 bits) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	4 bits

Description:

Spare 4 bits

Used by:

`AEMFRPWRREG (10.6.8)`

10.7.56 `TSP8` (Spare byte field) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`AemEnv0 (10.4.0)`

10.8 Algorithms

10.8.0 `LDTEMVADCCNV` (TEM Voltage Conversion) Algorithm

ITOS Mnemonic: `LDTEMVADCCNV`

Description:

Definition:

C0	0.00000000	C4	0.00000000
C1	0.00122100	C5	0.00000000
C2	0.00000000	C6	0.00000000
C3	0.00000000	C7	0.00000000

Used by:

`TEM33V (10.6.28)`

10.9 Discretes

10.9.0 `LAEMFRPWRSTATES` (AEM FREE Board Power States) Discrete

ITOS Mnemonic: `LAEMFRPWRSTATES`

Description:

Converts AEM FREE board power status values to ON/OFF discrete states.

Definition:

0 AEM Free Board Off State (`AEMFRBRDOFF`)
Denotes the OFF power state for an AEM free board

1 AEM FREE Board On State (`AEMFRBRDON`)
Denotes the ON power state for an AEM FREE board.

Used by:

`AEMFRPWRREG (10.6.8)`

10.9.1 `LAPDUPWRCNVTSTAT` (ACD PDU Power Converter State) Discrete

ITOS Mnemonic: `LAPDUPWRCNVTSTAT`

Description:

Definition:

0 ACD PDU Converter Primary (`LAPDUCNVTPRI`)

1 ACD PDU Power Converter Redundant (`LAPDUCNVTRED`)

Used by:

`ACDPWRREG (10.6.2)`

10.9.2 `LAPDUPWRSTATES` (ACD PDU Power States) Discrete

ITOS Mnemonic: `LAPDUPWRSTATES`

Description:

Definition:

0 ACD PDU Power Off State (`LAPDUPWROFFSTATE`)

1 ACD PDU Power On State (`LAPDUPWRONSTATE`)

Used by:

`ACDPWRREG (10.6.2)`

10.9.3 `LAPDUPWRSUPSTAT` (ACD PDU Power Supply Sources) Discrete

ITOS Mnemonic: `LAPDUPWRSUPSTAT`

Description:

Definition:

0 ACD PDU Power Supply Primary (LAPDUPWRSUPPRI)

1 ACD PDU Power Supply Redundant (LAPDUPWRSUPRED)

Used by:

ACDPWRREG (10.6.2)

10.9.4 LDPDUEPUCNVT (PDU EPU Converter) Discrete

ITOS Mnemonic: LDPDUEPUCNVT

Description:

Definition:

0 PDU EPU Converter Status Primary (LDPDUEPUCVTPRI)

1 PDU EPU Converter Status Redundant (LPDUEPUCVTRED)

Used by:

PDUEPUCRATEPWR (10.6.23)

10.9.5 LDPDUEPUPWRST (EPU Power States) Discrete

ITOS Mnemonic: LDPDUEPUPWRST

Description:

Definition:

0 PDU EPU Power Off State (LDPDUEPUPWROFF)

1 PDU EPU Power On State (LDPDUEPUPWRON)

Used by:

PDUEPUCRATEPWR (10.6.23)

10.9.6 LDPDUTEMPWRST (PDU TEM Power State) Discrete

ITOS Mnemonic: LDPDUTEMPWRST

Description:

Definition:

0 PDU TEM Power Off (LPDUTEMPWROFF)

1 PDU TEM Power On State (LPDUTEMPWRON)

Used by:

PDUTEMPWRREG (10.6.24)

10.9.7 LHKSTATUSBITS (ADC status bit conversions) Discrete

ITOS Mnemonic: LHKSTATUSBITS

Description:

Describes the discrete conversion of a 4 bit status value.

Definition:

0 Data OK status (LHKSTATOK)
Status indicating measurements was evaluated as OK.

1 Status Undefined (LHKSTATUNDF)
Indicates absence of data due to acquisition timeout or other failure.

2 Status Masked (LHKSTATMSK)
Indicates no evalutation status because value was masked as disabled.

3 ADC red limit status (LHKREDSTAT)
Status indicating an ADC red limit threshold violation.

Used by:

AEMFRHV1 (10.6.6), AEMFRHV2 (10.6.7), AEMFRTEMP (10.6.9), AEMFRVDD (10.6.10), EPUTEMP (10.6.19), EPUV (10.6.20), TEM33V (10.6.28), TEMPSTEMP (10.6.31), CALB-SPTEMP (10.6.16), TEMPCBTEMP (10.6.30), VCHPDSHPTEMP (10.6.39), VCHPRSVRHTRTEMP (10.6.40), VCHPXLHPTEMP (10.6.41), ACDBEAGRIDTEMP (10.6.0), ACDPMTRAILTEMP (10.6.1), ACDSHELLTEMP (10.6.3), GRIDRADIFTEMP (10.6.21), GRIDTEMP (10.6.22), RADANHTRTEMP (10.6.25), RADTEMP (10.6.26), CAL33I (10.6.12), CAL33V (10.6.13), CALBIASI (10.6.14), CALBIASV (10.6.15), TEM33I (10.6.27), TKR15I (10.6.32), TKR15V (10.6.33), TKR25I (10.6.34), TKR25V (10.6.35), TKRBIASI (10.6.36), TKR-BIASV (10.6.37), AFEETEMP (10.6.11), TKRCBLT (10.6.38)

10.9.8 LRLIMDEVICE (Device Opcodes for Red Limit Alerts) Discrete

ITOS Mnemonic: LRLIMDEVICE

Description:

Enumerates the device opcode that exceeded a red limit.

Definition:

0 AEM Free Board 0 (AEMFR0)

1 AEM Free Board 1 (AEMFR1)

2 AEM Free Board 2 (AEMFR2)

3 AEM Free Board 3 (AEMFR3)

4 AEM Free Board 4 (AEMFR4)

5 AEM Free Board 5 (`AEMFR5`)

6 AEM Free Board 6 (`AEMFR6`)

7 AEM Free Board 7 (`AEMFR7`)

8 AEM Free Board 8 (`AEMFR8`)

9 AEM Free Board 9 (`AEMFR9`)

10 AEM Free Board 10 (`AEMFR10`)

11 AEM Free Board 11 (`AEMFR11`)

12 AEM DAB (`AEMDAB`)

Used by:

`RedLimAlrt (10.4.46)`

10.10 Limit Sets

10.10.0 `LABEAGTEMPADCLIM` (ACD BEA Grid Interface Temperature Limits) Limit

ITOS Mnemonic: `LABEAGTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

???

10.10.1 `LAPMTRTEMPADCLIM` (ACD PMT Rail Temperature Limits) Limit

ITOS Mnemonic: `LAPMTRTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`ACDPMTRAILTEMP (10.6.1)`

10.10.2 `LASHLTEMPADCLIM` (ACD Shell Temperature Limits) Limit

ITOS Mnemonic: `LASHLTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`ACDSHELLTEMP (10.6.3)`

10.10.3 `LC33IADCLIM` (CAL 3.3I ADC limits) Limit

ITOS Mnemonic: `LC33IADCLIM`

Description:

Describes the CAL 3.3 current ADC limit set

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`CAL33I (10.6.12)`

10.10.4 `LC33VADCLIM` (CAL 3.3V ADC limits) Limit

ITOS Mnemonic: `LC33VADCLIM`

Description:

Describes the CAL 3.3 volt adc limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`CAL33V (10.6.13)`

10.10.5 `LCAFETADCLIM` (CAL AFFE temperature limits) Limit

ITOS Mnemonic: `LCAFETADCLIM`

Description:

Describes the CAL AFFE temperature limit set

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`AFEETEMP (10.6.11)`

10.10.6 `LCBASPLADCLIM` (CAL Baseplate Temperature Limits) Limit

ITOS Mnemonic: `LCBASPLADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`CALBSPTEMP (10.6.16)`

10.10.7 `LCBIASIADCLIM` (CAL bias current limits) Limit

ITOS Mnemonic: `LCBIASIADCLIM`

Description:

Describes the CAL bias current ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`CALBIASI (10.6.14)`

10.10.8 `LCBIASVADCLIM` (CAL bias voltage limits) Limit

ITOS Mnemonic: `LCBIASVADCLIM`

Description:

Describes the CAL bias voltage ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`CALBIASV (10.6.15)`

10.10.9 `LDAEMFRHV1ADCLIM` (AEM Free Board HV1 Limits) Limit

ITOS Mnemonic: `LDAEMFRHV1ADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047

Yellow alert	0 - 2047

Used by:

`AEMFRHV1 (10.6.6)`

10.10.10 **`LDAEMFRHV2ADCLIM`** (AEM Free Board HV2 Limits) Limit

ITOS Mnemonic: `LDAEMFRHV2ADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`AEMFRHV2 (10.6.7)`

10.10.11 **`LDAEMFRTMPADCLIM`** (AEM Free Board Temperature Limits) Limit

ITOS Mnemonic: `LDAEMFRTMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`AEMFRTEMP (10.6.9)`

10.10.12 **`LDAEMFRVDDADCLIM`** (AEM Free Board VDD Limits) Limit

ITOS Mnemonic: `LDAEMFRVDDADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`AEMFRVDD (10.6.10)`

10.10.13 `LDEPUTEMPADCLIM` (EPU Temperature ADC Limits) Limit

ITOS Mnemonic: `LDEPUTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`EPUTEMP (10.6.19)`

10.10.14 `LDEPUVADCLIM` (EPU Voltage ADC Limits) Limit

ITOS Mnemonic: `LDEPUVADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`EPUV (10.6.20)`

10.10.15 `LDTEM33IADCLIM` (TEM 3.3I digital limits) Limit

ITOS Mnemonic: `LDTEM33IADCLIM`

Description:

Describes the TEM 3.3 current ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TEM33I (10.6.27)`

10.10.16 `LDTEM33VADCLIM` (TEM digital 3.3V limits) Limit

ITOS Mnemonic: `LDTEM33VADCLIM`

Description:

Describes the TEM 3.3V digital ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TEM33V (10.6.28)`

10.10.17 `LDTEMPCBTADCLIM` (TEM PCB Temperature Limits) Limit

ITOS Mnemonic: `LDTEMPCBTADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TEMPCBTEMP (10.6.30)`

10.10.18 `LDTEMPSTADCLIM` (TEM Power Supply Temperature Limits) Limit

ITOS Mnemonic: `LDTEMPSTADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TEMPSTEMP (10.6.31)`

10.10.19 `LMGRDRADIFADCLIM` (Grid Radiator Interface Temperature Limits) Limit

ITOS Mnemonic: `LMGRDRADIFADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`GRIDRADIFTEMP (10.6.21)`

10.10.20 `LMGRIDTEMPADCLIM` (Grid Temperature ADC Limits) Limit

ITOS Mnemonic: `LMGRIDTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2048
Yellow alert	0 - 2047

Used by:

`GRIDTEMP (10.6.22)`

10.10.21 `LMRADAFHTRADCLIM` (Radiator Anitfreeze Heater Temperature Limits) Limit

ITOS Mnemonic: `LMRADAFHTRADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`RADANHTRTEMP (10.6.25)`

10.10.22 `LMRADTEMPADCLIM` (Radiator Temperature Limits) Limit

ITOS Mnemonic: `LMRADTEMPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`RADTEMP (10.6.26)`

10.10.23 `LMVCHPDSHPADCLIM` (VCHP-DSHP Interface Temperature Limits) Limit

ITOS Mnemonic: `LMVCHPDSHPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`VCHPDSHPTEMP (10.6.39)`

10.10.24 `LMVCHPRSVTADCLIM` (VCHP Reservoir Heater Temperature Limits) Limit

ITOS Mnemonic: `LMVCHPRSVTADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`VCHPRSVRHTRTEMP (10.6.40)`

10.10.25 `LMVCHPXLHPADCLIM` (VCHP-XLHP Interface Temperature Limits) Limit

ITOS Mnemonic: `LMVCHPXLHPADCLIM`

Description:

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`VCHPXLHPTEMP (10.6.41)`

10.10.26 `LT15IADCLIM` (TKR 1.5I ADC limits) Limit

ITOS Mnemonic: `LT15IADCLIM`

Description:

Describes the TKR 1.5I ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKR15I (10.6.32)`

10.10.27 `LT15VADCLIM` (TKR 1.5V ADC Limits) Limit

ITOS Mnemonic: `LT15VADCLIM`

Description:

Describes the limit set for a TKR 1.5 volt ADC value.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKR15V (10.6.33)`

10.10.28 `LT25IADCLIM` (TKR 2.5I ADC limits) Limit

ITOS Mnemonic: `LT25IADCLIM`

Description:

Describes the TKR 2.5I ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKR25I (10.6.34)`

10.10.29 `LT25VADCLIM` (TKR 2.5V ADC limits) Limit

ITOS Mnemonic: `LT25VADCLIM`

Description:

Describes the limits for a TKR 2.5 volt ADC value.

Definition:

Inversion flag	F

Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKR25V (10.6.35)`

10.10.30 `LTBIASIADCLIM` (TKR bias current ADC limits) Limit

ITOS Mnemonic: `LTBIASIADCLIM`

Description:

Describes the limit set for a TKR bias current ADC value.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKRBIASI (10.6.36)`

10.10.31 `LTBIASVADCLIM` (TKR bias voltage ADC limits) Limit

ITOS Mnemonic: `LTBIASVADCLIM`

Description:

Describes the limit set for a TKR bias voltage ADC value.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKRBIASV (10.6.37)`

10.10.32 `LTCBLTADCLIM` (TKR cable temperature limits) Limit

ITOS Mnemonic: `LTCBLTADCLIM`

Description:

Describes the TKR cable temperature ADC limit set.

Definition:

Inversion flag	F
Limit switch	none
Red alert	0 - 2047
Yellow alert	0 - 2047

Used by:

`TKRCBLT (10.6.38)`

11 LMC Package

11.0 Overview

The LMC package handles the LAT multiplexed counters.

11.1 Command Packets

ITOS Mnemonic: LMCACDTILEALL

11.1.0 `acd_tile_all` (1692/0x69C:3)

Context:

Description:

"ACD Tile Counters (All)" Telecommand Packet

Starts collection of all ACD tile counters.

Layout:

ACD Tile Counters (All) (acd_tile_all) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x69C:3										
002	SF=3		Sequence Count													
004	Packet Length=7															
006	0	Function Code=3														
008	interval: intv															
00A	count: cnt															
00C	Packet Checksum															

```
acd_tile_all (pkt)
|  Sample interval in milliseconds intv (fld)
|  Sample count cnt (fld)

typedef struct _LMC_acd_tile_all_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LMC_acd_tile_all_Prm pay;

} LMC_acd_tile_all_Pkt;
```

Fields:

`count` **(11.2.1)**

Attribute(s):

Brief: "Sample count"

Instance(s): `cnt`

`interval` **(11.2.3)**

Attribute(s):

Brief: "Sample interval in milliseconds"

Instance(s): `intv`

Time between read of counters in a duration.

ITOS Mnemonic: `LMCACDTILEPAIR`

11.1.1 `acd_tile_pair` (1692/0x69C:2)

Context:

Description:

"ACD Tile Counter (Pair)" Telecommand Packet

Starts collection of a pair of ACD tile counters.

Layout:

ACD Tile Counter (Pair) (`acd_tile_pair`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x69C:2**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=11**															
006	**0**	**Function Code=2**														
008	**interval:** `intv`															
00A	**count:** `cnt`															
00C	**tile_num:** `tile_0`															
00E	**tile_num:** `tile_1`															
010	**Packet Checksum**															

```
acd_tile_pair (pkt)
|  Sample interval in milliseconds intv (fld)
|  Sample count cnt (fld)
|  ACD Tile ID tile_0 (fld) ... tile_1 (fld)
```

```
typedef struct _LMC_acd_tile_pair_Pkt {
```

```
    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LMC_acd_tile_pair_Prm pay;

} LMC_acd_tile_pair_Pkt;
```

Fields:

`count` **(11.2.1)**

Attribute(s):

Brief: "Sample count"

Instance(s): `cnt`

`interval` **(11.2.3)**

Attribute(s):

Brief: "Sample interval in milliseconds"

Instance(s): `intv`

Time between read of counters in a duration.

`tile_num` **(11.2.5)**

Attribute(s):

Brief: "ACD Tile ID"

Instance(s): `tile_0`, `tile_1`

ITOS Mnemonic: `LMCCALLRS`

11.1.2 `cal_lrs` (1692/0x69C:0)

Context:

Description:

"CAL Low Rate Science Counters" Telecommand Packet

Start collection of CAL low rate science counter data

Layout:

CAL Low Rate Science Counters (cal_lrs) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x69C:0**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=13**															
006	**0**	**Function Code=0**														
008	**interval:** intv															
00A	**count:** cnt															
00C	**cal_mask:** cal_mask															
010	**tem_mask:** tem_mask															
012	**Packet Checksum**															

```
cal_lrs (pkt)
|  Sample interval in milliseconds intv (fld)
|  Sample count cnt (fld)
|  CAL LRS Mask cal_mask (fld)
|  TEM device mask tem_mask (fld)

typedef struct _LMC_cal_lrs_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LMC_cal_lrs_Prm      pay;

} LMC_cal_lrs_Pkt;
```

Fields:

`cal_mask` **(11.2.0)**

Attribute(s):

Brief: "CAL LRS Mask"

Instance(s): `cal_mask`

`count` **(11.2.1)**

Attribute(s):

Brief: "Sample count"

Instance(s): `cnt`

`interval` **(11.2.3)**

Attribute(s):

Brief: "Sample interval in milliseconds"

Instance(s): `intv`

Time between read of counters in a duration.

`tem_mask` **(11.2.4)**

Attribute(s):

Brief: "TEM device mask"

Instance(s): `tem_mask`

ITOS Mnemonic: `LMCSTOPCOUNT`

11.1.3 `stop_count` (1692/0x69C:4)

Context:

Description:

"Stop Active Counter" Telecommand Packet

Stop the active counter

Layout:

Stop Active Counter `(stop_count)`
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	**Version=0**			**T=1**	**SH**	**APID=0x69C:4**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=7**															
006	**0**	**Function Code=4**														
008	**counter_opcode:** `opcode`															
00C	**Packet Checksum**															

```
stop_count (pkt)
|  Counter type opcode opcode (fld)

typedef struct _LMC_stop_count_Pkt {

    /* CCSDS header */
    char                 hdr[8];
```

```
    /* Command payload */
    LMC_stop_count_Prm   pay;

} LMC_stop_count_Pkt;
```

Fields:

`counter_opcode` **(11.2.2)**

Attribute(s):

Brief: "Counter type opcode"

Instance(s): `opcode`

Designates the counter type CAL, TKR, or ACD

ITOS Mnemonic: `LMCTKRLRS`

11.1.4 `tkr_lrs` (1692/0x69C:1)

Context:

Description:

"TKR Low Rate Science Counters" Telecommand Packet

Starts collection of TKR 3-in-a-row counters.

Layout:

TKR Low Rate Science Counters (`tkr_lrs`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x69C:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=11**															
006	**0**	**Function Code=1**														
008	**interval:** `intv`															
00A	**count:** `cnt`															
00C	**tkr_mask:** `tkr_mask`															
00E	**tem_mask:** `tem_mask`															
010	**Packet Checksum**															

```
tkr_lrs (pkt)
|  Sample interval in milliseconds intv (fld)
|  Sample count cnt (fld)
|  TKR LRS Mask tkr_mask (fld)
|  TEM device mask tem_mask (fld)
```

```
typedef struct _LMC_tkr_lrs_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LMC_tkr_lrs_Prm      pay;

} LMC_tkr_lrs_Pkt;
```

Fields:

`count` **(11.2.1)**

Attribute(s):

Brief: "Sample count"

Instance(s): `cnt`

`interval` **(11.2.3)**

Attribute(s):

Brief: "Sample interval in milliseconds"

Instance(s): `intv`

Time between read of counters in a duration.

`tem_mask` **(11.2.4)**

Attribute(s):

Brief: "TEM device mask"

Instance(s): `tem_mask`

`tkr_mask` **(11.2.6)**

Attribute(s):

Brief: "TKR LRS Mask"

Instance(s): `tkr_mask`

11.2 Command Fields

11.2.0 `cal_mask` (CAL LRS Mask) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`cal_lrs (11.1.2)`

11.2.1 `count` (Sample count) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`acd_tile_all (11.1.0)`

11.2.2 `counter_opcode` (Counter type opcode) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Designates the counter type CAL, TKR, or ACD

Used by:

`stop_count (11.1.3)`

11.2.3 `interval` (Sample interval in milliseconds) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Time between read of counters in a duration.

Used by:

`acd_tile_all (11.1.0)`

11.2.4 `tem_mask` (TEM device mask) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`cal_lrs (11.1.2)`

11.2.5 `tile_num` (ACD Tile ID) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`acd_tile_pair (11.1.1)`

11.2.6 `tkr_mask` (TKR LRS Mask) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`tkr_lrs (11.1.4)`

11.3 Telemetry Packets

11.3.0 `acd_cnt` (707/0x2C3)

Context:

Description:

"ACD Tile Counters" Telemetry Packet

Layout:

<table>
<tr><th colspan="17">ACD Tile Counters (acd_cnt)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x2C3</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=333</td></tr>
<tr><td>006</td><td colspan="16">Timestamp</td></tr>
<tr><td>00E</td><td colspan="16">spare16: spare</td></tr>
<tr><td>010</td><td colspan="16">tile_counter: cntrs[0-26]</td></tr>
</table>

```
acd_cnt (pkt)
|  Spare 16 bits spare (fld)
|  ADC Tile Counter Structure cntrs[0-26] (str)
|  |  ACD Tile ID tile_0 (fld) ... tile_1 (fld)
|  |  Delta timestamp dt (fld)
|  |  Counter data cnt (fld)

typedef struct _LMC_acd_cnt_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Spare 16 bits */
    unsigned short       spare;

    /* ADC Tile Counter Structure */
    LMC_tile_counter     cntrs[27];

} LMC_acd_cnt_Tlm;
```

Fields:

`spare16` **(11.5.4)**

Attribute(s):

Brief: "Spare 16 bits"

Instance(s): `spare`

`tile_counter` **(11.4.1)**

Attribute(s):

Brief: "ADC Tile Counter Structure"

Instance(s): `cntrs`

11.3.1 `cal_cnt` (705/0x2C1)

Context:

Description:

"CAL Low Rate Science Counters" Telemetry Packet

Layout:

CAL Low Rate Science Counters (cal_cnt) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x2C1**										
002	**SF**		**Sequence Count**													
004	**Packet Length=141**															
006	**Timestamp**															
00E	**dev_mask:** dmsk															
010	**lrs_mask:** lmsk															
014	**cal_counter:** cntrs[0-15]															

```
cal_cnt (pkt)
|  TEM mask dmsk (fld)
|  Low Rate Science Mask lmsk (fld)
|  CAL LRS Counter Structure cntrs[0-15] (str)
|  |  Delta timestamp dt (fld)
|  |  Counter data cnt (fld)

typedef struct _LMC_cal_cnt_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* TEM mask */
    unsigned short       dmsk;

    /* Low Rate Science Mask */
    unsigned int         lmsk;
```

```
    /* CAL LRS Counter Structure */
    LMC_cal_counter      cntrs[16];

} LMC_cal_cnt_Tlm;
```

Fields:

`cal_counter` **(11.4.0)**

Attribute(s):

Brief: "CAL LRS Counter Structure"

Instance(s): `cntrs`

`dev_mask` **(11.5.1)**

Attribute(s):

Brief: "TEM mask"

Instance(s): `dmsk`

`lrs_mask` **(11.5.3)**

Attribute(s):

Brief: "Low Rate Science Mask"

Instance(s): `lmsk`

11.3.2 `tkr_cnt` (706/0x2C2)

Context:

Description:

"TKR Low Rate Science Counters" Telemetry Packet

Layout:

<table>
<tr><th colspan="17">TKR Low Rate Science Counters (tkr_cnt)
Telemetry Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=0</td><td>SH</td><td colspan="11">APID=0x2C2</td></tr>
<tr><td>002</td><td colspan="2">SF</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=269</td></tr>
</table>

TKR Low Rate Science Counters (tkr_cnt) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
006	**Timestamp**															
00E	**dev_mask:** dmsk															
010	**lrs_mask:** lmsk															
014	**tkr_counter:** cntrs[0-31]															

```
tkr_cnt (pkt)
|  TEM mask dmsk (fld)
|  Low Rate Science Mask lmsk (fld)
|  TKR LRS Counter Structure cntrs[0-31] (str)
|  |  Delta timestamp dt (fld)
|  |  Counter data cnt (fld)

typedef struct _LMC_tkr_cnt_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* TEM mask */
    unsigned short       dmsk;

    /* Low Rate Science Mask */
    unsigned int         lmsk;

    /* TKR LRS Counter Structure */
    LMC_tkr_counter      cntrs[32];

} LMC_tkr_cnt_Tlm;
```

Fields:

`dev_mask` **(11.5.1)**

Attribute(s):

Brief: "TEM mask"

Instance(s): `dmsk`

`lrs_mask` **(11.5.3)**

Attribute(s):

Brief: "Low Rate Science Mask"

Instance(s): `lmsk`

`tkr_counter` **(11.4.2)**

Attribute(s):

Brief: "TKR LRS Counter Structure"

Instance(s): `cntrs`

11.4 Telemetry Structs

11.4.0 `cal_counter` (CAL LRS Counter Structure) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 64 bits (8 bytes)

Description:

CAL LRS Counter Structure (`cal_counter`) Telemetry Field Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	**dtime:** `dt`															
004	**counter:** `cnt`															

```
cal_counter (str)
|  Delta timestamp dt (fld)
|  Counter data cnt (fld)
```

```
typedef struct _LMC_cal_counter {

    /* Delta timestamp */
    unsigned int         dt;

    /* Counter data */
    unsigned int         cnt;

} LMC_cal_counter;
```

Fields:

`counter` **(11.5.0)**

Attribute(s):

Brief: "Counter data"

Instance(s): `cnt`

Variable data structure

`dtime` **(11.5.2)**

Attribute(s):

Brief: "Delta timestamp"

Instance(s): `dt`

Same as:

`tkr_counter (11.4.2)`

Used by:

`cal_cnt (11.3.1)`

11.4.1 `tile_counter` (ADC Tile Counter Structure) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 96 bits (12 bytes)

Description:

ADC Tile Counter Structure (`tile_counter`) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**tile_id:** `tile_0`															
002	**tile_id:** `tile_1`															
004	**dtime:** `dt`															
008	**counter:** `cnt`															

```
tile_counter (str)
|  ACD Tile ID tile_0 (fld) ... tile_1 (fld)
|  Delta timestamp dt (fld)
|  Counter data cnt (fld)
```

```
typedef struct _LMC_tile_counter {

    /* ACD Tile ID */
    unsigned short       tile_0;

    /* ACD Tile ID */
    unsigned short       tile_1;

    /* Delta timestamp */
    unsigned int         dt;

    /* Counter data */
    unsigned int         cnt;

} LMC_tile_counter;
```

Fields:

`counter` **(11.5.0)**

Attribute(s):

Brief: "Counter data"

Instance(s): `cnt`

Variable data structure

`dtime` **(11.5.2)**

Attribute(s):

Brief: "Delta timestamp"

Instance(s): `dt`

`tile_id` **(11.5.5)**

Attribute(s):

Brief: "ACD Tile ID"

Instance(s): `tile_0`, `tile_1`

Used by:

`acd_cnt (11.3.0)`

11.4.2 `tkr_counter` (TKR LRS Counter Structure) Telemetry Struct

Definition:

Alignment: 4 bytes
Length: 64 bits (8 bytes)

Description:

TKR LRS Counter Structure (`tkr_counter`) Telemetry Field Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**dtime:** `dt`															
004	**counter:** `cnt`															

```
tkr_counter (str)
|  Delta timestamp dt (fld)
|  Counter data cnt (fld)
```

```
typedef struct _LMC_tkr_counter {

    /* Delta timestamp */
    unsigned int         dt;

    /* Counter data */
    unsigned int         cnt;

} LMC_tkr_counter;
```

Fields:

`counter` **(11.5.0)**

Attribute(s):

Brief: "Counter data"

Instance(s): `cnt`

Variable data structure

`dtime` **(11.5.2)**

Attribute(s):

Brief: "Delta timestamp"

Instance(s): `dt`

Same as:

`cal_counter (11.4.0)`

Used by:

`tkr_cnt (11.3.2)`

11.5 Telemetry Fields

11.5.0 `counter` (Counter data) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Variable data structure

Used by:

`tile_counter (11.4.1)`

11.5.1 `dev_mask` (TEM mask) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`cal_cnt (11.3.1)`

11.5.2 `dtime` (Delta timestamp) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`tile_counter (11.4.1)`

11.5.3 `lrs_mask` (Low Rate Science Mask) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`cal_cnt (11.3.1)`

11.5.4 `spare16` (Spare 16 bits) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`acd_cnt (11.3.0)`

11.5.5 `tile_id` (ACD Tile ID) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`tile_counter (11.4.1)`

12 LSM Package

12.0 Overview

The LSM package processes the seven telecommands per second that the spacecraft sends the LAT continuously to inform the LAT of things like orbital position, time, etc. LSM runs in its own task.

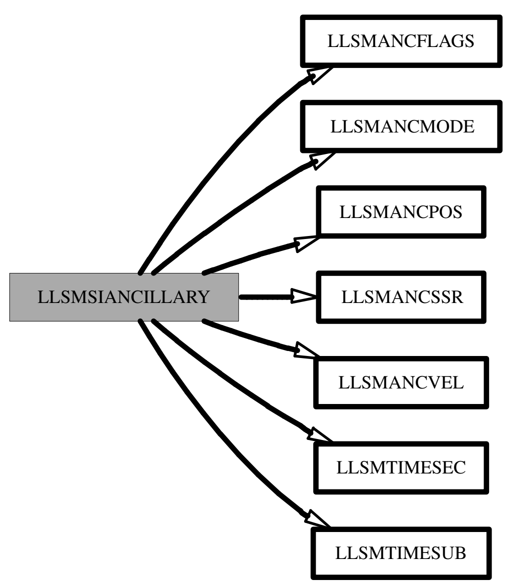
The package supports the following functions:

- Wall clock time services (GPS)

12.1 Command Packets

12.1.0 `LLSMSIANCILLARY` (1793/0x701:2)

Context:



Description:

"SC Ancillary Broadcast Message" Telecommand Packet

Layout:

SC Ancillary Broadcast Message (LLSMSIANCILLARY) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x701:2										
002	SF=3		Sequence Count													

SC Ancillary Broadcast Message (LLSMSIANCILLARY) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
004	**Packet Length=39**															
006	**0**	**Function Code=2**														
008	**LLSMTIMESEC:** sec															
00C	**LLSMTIMESUB:** sub															
010	**LLSMANCPOS:** pos[0-2]															
01C	**LLSMANCVEL:** vel[0-2]															
028	**LLSMANCMODE:** mode								**LLSMANCSSR:** ssr							
02A	**LLSMANCFLAGS:** flags															
02C	**Packet Checksum**															

```
LLSMSIANCILLARY (pkt)
|  Timestamp seconds sec (fld)
|  Timestamp microseconds sub (fld)
|  Ancillary position element pos[0-2] (fld)
|  Ancillary velocity element vel[0-2] (fld)
|  Ancillary GNC mode mode (fld)
|  Ancillary SSR usage ssr (fld)
|  Ancillary SC flags flags (fld)

typedef struct _LSM_LLSMSIANCILLARY_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LSM_LLSMSIANCILLARY_Prm pay;

} LSM_LLSMSIANCILLARY_Pkt;
```

Fields:

`LLSMANCFLAGS` **(12.2.0)**

Attribute(s):

Brief: "Ancillary SC flags"

Instance(s): `flags`

`LLSMANCMODE` **(12.2.1)**

Attribute(s):

Brief: "Ancillary GNC mode"

Instance(s): `mode`

`LLSMANCPOS` **(12.2.2)**

Attribute(s):

Brief: "Ancillary position element"

Instance(s): `pos`

`LLSMANCSSR` **(12.2.3)**

Attribute(s):

Brief: "Ancillary SSR usage"

Instance(s): `ssr`

`LLSMANCVEL` **(12.2.4)**

Attribute(s):

Brief: "Ancillary velocity element"

Instance(s): `vel`

`LLSMTIMESEC` **(12.2.8)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `sec`

`LLSMTIMESUB` **(12.2.9)**

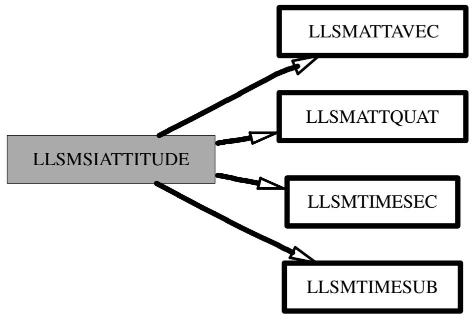
Attribute(s):

Brief: "Timestamp microseconds"

Instance(s): `sub`

12.1.1 **LLSMSIATTITUDE** (1793/0x701:1)

Context:



Description:

"SC Attitude Broadcast Message" Telecommand Packet

The SC sends this message 5 times a second on the CTDB bus. It contains information about the SC-J2000 attitude.

Layout:

SC Attitude Broadcast Message (LLSMSIATTITUDE) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x701:1										
002	SF=3		Sequence Count													
004	Packet Length=55															
006	0	Function Code=1														
008	LLSMTIMESEC: sec															
00C	LLSMTIMESUB: sub															
010	LLSMATTQUAT: quat[0-3]															
030	LLSMATTAVEC: avel[0-2]															
03C	Packet Checksum															

```
LLSMSIATTITUDE (pkt)
|  Timestamp seconds sec (fld)
|  Timestamp microseconds sub (fld)
|  Attitude quaternion element quat[0-3] (fld)
|  Attitude angular velocity element avel[0-2] (fld)

typedef struct _LSM_LLSMSIATTITUDE_Pkt {

    /* CCSDS header */
```

```
    char                 hdr[8];

    /* Command payload */
    LSM_LLSMSIATTITUDE_Prm pay;

} LSM_LLSMSIATTITUDE_Pkt;
```

Fields:

`LLSMATTAVEC` **(12.2.5)**

Attribute(s):

Brief: "Attitude angular velocity element"

Instance(s): `avel`

`LLSMATTQUAT` **(12.2.6)**

Attribute(s):

Brief: "Attitude quaternion element"

Instance(s): `quat`

`LLSMTIMESEC` **(12.2.8)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `sec`

`LLSMTIMESUB` **(12.2.9)**

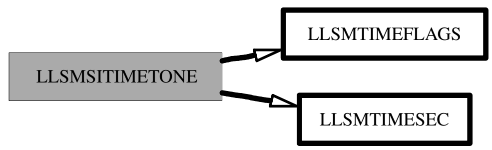
Attribute(s):

Brief: "Timestamp microseconds"

Instance(s): `sub`

12.1.2 `LLSMSITIMETONE` (1793/0x701:3)

Context:



Description:

"SC Timetone Broadcast Message" Telecommand Packet

Layout:

SC Timetone Broadcast Message (`LLSMSITIMETONE`) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x701:3**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=9**															
006	**0**	**Function Code=3**														
008	**LLSMTIMESEC:** `sec`															
00C	**LLSMTIMEFLAGS:** `flags`															
00E	**Packet Checksum**															

```
LLSMSITIMETONE (pkt)
|  Timestamp seconds sec (fld)
|  Timetone SC flags flags (fld)

typedef struct _LSM_LLSMSITIMETONE_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LSM_LLSMSITIMETONE_Prm pay;

} LSM_LLSMSITIMETONE_Pkt;
```

Fields:

`LLSMTIMEFLAGS` **(12.2.7)**

Attribute(s):

Brief: "Timetone SC flags"

Instance(s): `flags`

`LLSMTIMESEC` **(12.2.8)**

Attribute(s):

Brief: "Timestamp seconds"

Instance(s): `sec`

12.2 Command Fields

12.2.0 `LLSMANCFLAGS` (Ancillary SC flags) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.1 `LLSMANCMODE` (Ancillary GNC mode) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.2 `LLSMANCPOS` (Ancillary position element) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	float
Length:	32 bits (4 bytes)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.3 `LLSMANCSSR` (Ancillary SSR usage) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.4 `LLSMANCVEL` (Ancillary velocity element) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	float
Length:	32 bits (4 bytes)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.5 `LLSMATTAVEC` (Attitude angular velocity element) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	float
Length:	32 bits (4 bytes)

Description:

Used by:

`LLSMSIATTITUDE (12.1.1)`

12.2.6 `LLSMATTQUAT` (Attitude quaternion element) Telecommand Field

Definition:

Alignment:	8 bytes
C type:	double
Length:	64 bits (8 bytes)

Description:

Used by:

`LLSMSIATTITUDE (12.1.1)`

12.2.7 `LLSMTIMEFLAGS` (Timetone SC flags) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LLSMSITIMETONE (12.1.2)`

12.2.8 `LLSMTIMESEC` (Timestamp seconds) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

12.2.9 `LLSMTIMESUB` (Timestamp microseconds) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`LLSMSIANCILLARY (12.1.0)`

13 LTC Package

13.0 Overview

The LTC package provides the LAT thermal control mechanism. It reads temperatures from within the instrument, and on that basis, decides how much heat to dump. LTC runs as its own task.

13.1 Command Packets

ITOS Mnemonic: `LTCHtrOnOffCtl`

13.1.0 `HtrOnOffCtl` (1605/0x645:5)

Context:

Description:

"Set heater to always on, or off or automatic control." Telecommand Packet

Sets a specified heater to on, or off or automatic control.

Layout:

<table>
<tr><th colspan="17">Set heater to always on, or off or automatic control. (HtrOnOffCtl)
Telecommand Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=1</td><td>SH</td><td colspan="11">APID=0x645:5</td></tr>
<tr><td>002</td><td colspan="2">SF=3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=7</td></tr>
<tr><td>006</td><td>0</td><td colspan="15">Function Code=5</td></tr>
<tr><td>008</td><td colspan="16">WORD16: HtrNum</td></tr>
<tr><td>00A</td><td colspan="16">WORD16: OnOffCtl</td></tr>
<tr><td>00C</td><td colspan="16">Packet Checksum</td></tr>
</table>

```
HtrOnOffCtl (pkt)
│  Unsigned 16-bit word HtrNum (fld) ... OnOffCtl (fld)

typedef struct _LTC_HtrOnOffCtl_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_HtrOnOffCtl_Prm  pay;

} LTC_HtrOnOffCtl_Pkt;
```

Fields:

`WORD16` **(13.2.2)**

Attribute(s):

Brief: "Unsigned 16-bit word"

Instance(s): `HtrNum`, `OnOffCtl`

ITOS Mnemonic: `LTCReStart`

13.1.1 `ReStart` (1605/0x645:1)

Context:

Description:

"Restart and initialize Thermal Control" Telecommand Packet

Restart thermal control by reading configuration files and initializing. Depending on paramter may start active control or passive operation.

Layout:

Restart and initialize Thermal Control (`ReStart`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x645:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=11**															
006	**0**	**Function Code=1**														
008	**FileId:** `cnfgFileId`															
00C	**FileId:** `snsrFileId`															
010	**Packet Checksum**															

```
ReStart (pkt)
|  File identification number cnfgFileId (fld) ... snsrFileId (fld)

typedef struct _LTC_ReStart_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_ReStart_Prm      pay;

} LTC_ReStart_Pkt;
```

Fields:

`FileId` **(13.2.1)**

Attribute(s):

Brief: "File identification number"

Instance(s): `cnfgFileId`, `snsrFileId`

ITOS Mnemonic: `LTCSetMode`

13.1.2 `SetMode` (1605/0x645:4)

Context:

Description:

"Set thermal control mode to active or passive." Telecommand Packet

Sets thermal control processing to active or processing. Active is normal control by turning on or off heat pipe reservoir heaters to keep termperature within specified limits. Passive does all normal control processing, but does not send commands to heaters.

Layout:

<table>
<tr><th colspan="17">Set thermal control mode to active or passive. (SetMode)
Telecommand Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=1</td><td>SH</td><td colspan="11">APID=0x645:4</td></tr>
<tr><td>002</td><td colspan="2">SF=3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=5</td></tr>
<tr><td>006</td><td>0</td><td colspan="15">Function Code=4</td></tr>
<tr><td>008</td><td colspan="16">WORD16: ActiveOrPassive</td></tr>
<tr><td>00A</td><td colspan="16">Packet Checksum</td></tr>
</table>

```
SetMode (pkt)
|  Unsigned 16-bit word ActiveOrPassive (fld)

typedef struct _LTC_SetMode_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_SetMode_Prm      pay;

} LTC_SetMode_Pkt;
```

Fields:

`WORD16` **(13.2.2)**

Attribute(s):

Brief: "Unsigned 16-bit word"

Instance(s): `ActiveOrPassive`

ITOS Mnemonic: `LTCSetParam`

13.1.3 `SetParam` (1605/0x645:6)

Context:

Description:

"Set control parameters to new values." Telecommand Packet

Sets reservoir and RIT temperature control limits to new values for all or a specified heat pipe. New values are used immediately.

Layout:

Set control parameters to new values. (SetParam)

Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x645:6										
002	SF=3		Sequence Count													
004	Packet Length=27															
006	0	Function Code=6														
008	WORD16: HtPipeSel															
00C	FLOAT: ResLoLimit															
010	FLOAT: ResHiLimit															
014	FLOAT: RitLoLimit															
018	FLOAT: RitHiLimit															
01C	FLOAT: RitDbDelta															
020	Packet Checksum															

```
SetParam (pkt)
|  Unsigned 16-bit word HtPipeSel (fld)
|  32-bit real number. ResLoLimit (fld) ... RitDbDelta (fld)

typedef struct _LTC_SetParam_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_SetParam_Prm     pay;

} LTC_SetParam_Pkt;
```

Fields:

`FLOAT` **(13.2.0)**

Attribute(s):

Brief: "32-bit real number."

Instance(s): `ResHiLimit`, `ResLoLimit`, `RitDbDelta`, `RitHiLimit`, `RitLoLimit`

`WORD16` **(13.2.2)**

Attribute(s):

Brief: "Unsigned 16-bit word"

Instance(s): `HtPipeSel`

ITOS Mnemonic: `LTCSetTlmFreq`

13.1.4 `SetTlmFreq` (1605/0x645:7)

Context:

Description:

"Set LTC telemetry frequency, 0 is off." Telecommand Packet

Layout:

Set LTC telemetry frequency, 0 is off. (`SetTlmFreq`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x645:7**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=5**															
006	**0**	**Function Code=7**														
008	**WORD16:** `TlmFreq`															
00A	**Packet Checksum**															

```
SetTlmFreq (pkt)
│  Unsigned 16-bit word TlmFreq (fld)

typedef struct _LTC_SetTlmFreq_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_SetTlmFreq_Prm   pay;

} LTC_SetTlmFreq_Pkt;
```

Fields:

`WORD16` **(13.2.2)**

Attribute(s):

Brief: "Unsigned 16-bit word"

Instance(s): `TlmFreq`

ITOS Mnemonic: `LTCStart`

13.1.5 `Start` (1605/0x645:2)

Context:

Description:

"Start control" Telecommand Packet

Start active or passive control.

Layout:

Start control (`Start`) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x645:2										
002	SF=3		Sequence Count													
004	Packet Length=5															
006	0	Function Code=2														
008	WORD16: `ActiveOrPassive`															
00A	Packet Checksum															

```
Start (pkt)
|  Unsigned 16-bit word ActiveOrPassive (fld)

typedef struct _LTC_Start_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    LTC_Start_Prm        pay;

} LTC_Start_Pkt;
```

Fields:

`WORD16` **(13.2.2)**

Attribute(s):

Brief: "Unsigned 16-bit word"

Instance(s): `ActiveOrPassive`

ITOS Mnemonic: `LTCStop`

13.1.6 `Stop` (1605/0x645:3)

Context:

Description:

"Terminate Thermal Control processing" Telecommand Packet

Terminates thermal control processing tasks, but message processing remains.

Layout:

Terminate Thermal Control processing (Stop) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x645:3										
002	SF=3		Sequence Count													
004	Packet Length=3															
006	0	Function Code=3														

```
Stop (pkt)

typedef struct _LTC_Stop_Pkt {

    /* CCSDS header */
    char                 hdr[8];

} LTC_Stop_Pkt;
```

13.2 Command Fields

13.2.0 `FLOAT` (32-bit real number.) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	float
Length:	32 bits (4 bytes)

Description:

Used by:

`SetParam (13.1.3)`

13.2.1 `FileId` (File identification number) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Used by:

`ReStart (13.1.1)`

13.2.2 `WORD16` (Unsigned 16-bit word) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`HtrOnOffCtl (13.1.0)`

13.3 Telemetry Packets

13.3.0 `DiagLTC` (700/0x2BC)

Context:

Description:

"LAT Thermal Control diagnostic telemetry" Telemetry Packet

Layout:

LAT Thermal Control diagnostic telemetry `(DiagLTC)` **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x2BC**										
002	**SF**		**Sequence Count**													
004	**Packet Length=240**															
006	**Timestamp**															
00E	**ResRitSnsrSel:** `snsrSel[0-11]`															
026	**ResRitStatus:** `snsrStatus[0-11]`															
032	**ActiveHtPipe:** `actvHp`															
034	**RawAdcStatVal:** `rawAdc[0-71]`															
0C4	**TempCelsius:** `temps[0-23]`															
0F4	**HeaterCmdMask:** `htrCmd`															

```
DiagLTC (pkt)
|  Selected RSVR and RIT sensors snsrSel[0-11] (bf)
|  |  Specifies connected HP number for sensor resHpNum (0-3)
|  |  Specifies Sensor Type resType (4-7)
|  |  Specifies connected HP number for sensor ritHpNum (8-11)
|  |  Specifies Sensor Type ritType (12-15)
|  Heat pipe RIT and reservoir sensors status snsrStatus[0-11] (bf)
|  |  Status for HP sensors ritStatus (0-3) ... resStatus (4-7)
|  Mask for active heat pipes actvHp (fld)
|  Raw ADC Status and Value rawAdc[0-71] (fld)
|  Temperature converted to celsius. temps[0-23] (fld)
|  Command mask for reservoir heaters htrCmd (fld)

typedef struct _LTC_DiagLTC_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Selected RSVR and RIT sensors */
    LTC_ResRitSnsrSel    snsrSel[12];

    /* Heat pipe RIT and reservoir sensors status */
    LTC_ResRitStatus     snsrStatus[12];
```

```
    /* Byte pad */
    char                 byte_pad0;

    /* Mask for active heat pipes */
    unsigned short       actvHp;

    /* Raw ADC Status and Value */
    unsigned short       rawAdc[72];

    /* Temperature converted to celsius. */
    short                temps[24];

    /* Command mask for reservoir heaters */
    unsigned short       htrCmd;

} LTC_DiagLTC_Tlm;
```

Fields:

`ActiveHtPipe` **(13.5.0)**

Attribute(s):

Brief: "Mask for active heat pipes"

Instance(s): `actvHp`

1 - Active 0 - Not active (used)

`HeaterCmdMask` **(13.5.1)**

Attribute(s):

Brief: "Command mask for reservoir heaters"

Instance(s): `htrCmd`

1 - turn on, and 0 turn off. This is opposite of the Heater Control Register.

`RawAdcStatVal` **(13.5.2)**

Attribute(s):

Brief: "Raw ADC Status and Value"

Instance(s): `rawAdc`

Upper 4 bits status Lower 12 bits raw value as read from PDU.

`ResRitSnsrSel` **(13.4.0)**

Attribute(s):

Brief: "Selected RSVR and RIT sensors"

Instance(s): `snsrSel`

`ResRitStatus` **(13.4.1)**

Attribute(s):

Brief: "Heat pipe RIT and reservoir sensors status"

Instance(s): `snsrStatus`

`TempCelsius` **(13.5.6)**

Attribute(s):

Brief: "Temperature converted to celsius."

Instance(s): `temps`

Fixed point scaled by 8 bits (256).

13.4 Telemetry Bitfields

13.4.0 `ResRitSnsrSel` (Selected RSVR and RIT sensors) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	`SensorHpNum resHpNum`
4	7	4	`SensorType resType`
8	11	4	`SensorHpNum ritHpNum`
12	15	4	`SensorType ritType`

```
ResRitSnsrSel (bf)
|  Specifies connected HP number for sensor resHpNum (0-3)
|  Specifies Sensor Type resType (4-7)
|  Specifies connected HP number for sensor ritHpNum (8-11)
|  Specifies Sensor Type ritType (12-15)
```

```
typedef struct _LTC_ResRitSnsrSel_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Specifies connected HP number for sensor */
    unsigned short       resHpNum:4;

    /* Specifies Sensor Type */
    unsigned short       resType:4;

    /* Specifies connected HP number for sensor */
    unsigned short       ritHpNum:4;

    /* Specifies Sensor Type */
    unsigned short       ritType:4;

    #elif ENDIANNESS_IS_LITTLE

    /* Specifies Sensor Type */
    unsigned short       ritType:4;

    /* Specifies connected HP number for sensor */
    unsigned short       ritHpNum:4;

    /* Specifies Sensor Type */
    unsigned short       resType:4;

    /* Specifies connected HP number for sensor */
    unsigned short       resHpNum:4;
```

```
    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LTC_ResRitSnsrSel_Bfl;

typedef union _LTC_ResRitSnsrSel {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _LTC_ResRitSnsrSel_Bfl   bf;

} LTC_ResRitSnsrSel;
```

Fields:

`SensorHpNum` **(13.5.3)**

Attribute(s):

Brief: "Specifies connected HP number for sensor"

Instance(s): `resHpNum`, `ritHpNum`

`SensorType` **(13.5.5)**

Attribute(s):

Brief: "Specifies Sensor Type"

Instance(s): `resType`, `ritType`

0-DSHP, 1-DSHP Redundant, 2-XLHP, 3-XLHP Redundant, 4-XLHP, 5-XLHP Redundant

Used by:

`DiagLTC (13.3.0)`

13.4.1 `ResRitStatus` (Heat pipe RIT and reservoir sensors status) Telemetry Bitfield

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Start	Stop	Size	Item Type and Name
0	3	4	`SensorStatus ritStatus`
4	7	4	`SensorStatus resStatus`

```
ResRitStatus (bf)
|  Status for HP sensors ritStatus (0-3) ... resStatus (4-7)
```

```
typedef struct _LTC_ResRitStatus_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Status for HP sensors */
    unsigned char        ritStatus:4;

    /* Status for HP sensors */
    unsigned char        resStatus:4;

    #elif ENDIANNESS_IS_LITTLE

    /* Status for HP sensors */
    unsigned char        resStatus:4;

    /* Status for HP sensors */
    unsigned char        ritStatus:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) LTC_ResRitStatus_Bfl;

typedef union _LTC_ResRitStatus {

    /* underlying unsigned char */
    unsigned char                   uc;

    /* struct mapped bitfield */
    struct _LTC_ResRitStatus_Bfl    bf;

} LTC_ResRitStatus;
```

Fields:

`SensorStatus` **(13.5.4)**

Attribute(s):

Brief: "Status for HP sensors"

Instance(s): `resStatus`, `ritStatus`

0-NOT_READ, 1-READ_FAIL, 2-GOOD, 3-BROKEN Use for RIT and reservoir sensor status for each heat pipe.

Used by:

`DiagLTC (13.3.0)`

13.5 Telemetry Fields

13.5.0 `ActiveHtPipe` (Mask for active heat pipes) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

1 - Active 0 - Not active (used)

Used by:

`DiagLTC (13.3.0)`

13.5.1 `HeaterCmdMask` (Command mask for reservoir heaters) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

1 - turn on, and 0 turn off. This is opposite of the Heater Control Register.

Used by:

`DiagLTC (13.3.0)`

13.5.2 `RawAdcStatVal` (Raw ADC Status and Value) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Upper 4 bits status Lower 12 bits raw value as read from PDU.

Used by:

`DiagLTC (13.3.0)`

13.5.3 `SensorHpNum` (Specifies connected HP number for sensor) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	4 bits

Description:

Used by:

`ResRitSnsrSel (13.4.0)`

13.5.4 `SensorStatus` (Status for HP sensors) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	4 bits

Description:

0-NOT_READ, 1-READ_FAIL, 2-GOOD, 3-BROKEN Use for RIT and reservoir sensor status for each heat pipe.

Used by:

`ResRitStatus (13.4.1)`

13.5.5 `SensorType` (Specifies Sensor Type) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	4 bits

Description:

0-DSHP, 1-DSHP Redundant, 2-XLHP, 3-XLHP Redundant, 4-XLHP, 5-XLHP Redundant

Used by:

`ResRitSnsrSel (13.4.0)`

13.5.6 `TempCelsius` (Temperature converted to celsius.) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	short
Length:	16 bits (2 bytes)

Description:

Fixed point scaled by 8 bits (256).

Used by:

`DiagLTC (13.3.0)`

14 MEM Package

14.0 Overview

The MEM package contains routines that are specific to the Memory Dump/Load facility.

14.1 Command Packets

ITOS Mnemonic: `LMEMDUMPCANCEL`

14.1.0 `LMEMDUMPCANCEL` (1604/0x644:1)

Context:

Description:

"Memory Dump Cancel" Telecommand Packet

Cancel a memory dump operation.

Layout:

<table>
<tr><th colspan="17">Memory Dump Cancel (LMEMDUMPCANCEL)
Telecommand Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=1</td><td>SH</td><td colspan="11">APID=0x644:1</td></tr>
<tr><td>002</td><td colspan="2">SF=3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=5</td></tr>
<tr><td>006</td><td>0</td><td colspan="15">Function Code=1</td></tr>
<tr><td>008</td><td colspan="16">CIDBF: idents</td></tr>
<tr><td>00A</td><td colspan="16">Packet Checksum</td></tr>
</table>

```
LMEMDUMPCANCEL (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)

typedef struct _MEM_LMEMDUMPCANCEL_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPCANCEL_Prm pay;

} MEM_LMEMDUMPCANCEL_Pkt;
```

Fields:

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

ITOS Mnemonic: `LMEMDUMPMEM`

14.1.1 `LMEMDUMPMEM` (1604/0x644:0)

Context:

Description:

"Memory Data Dump" Telecommand Packet

Dump data from a region of memory.

Layout:

Memory Data Dump (LMEMDUMPMEM) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x644:0										
002	SF=3		Sequence Count													
004	Packet Length=15															
006	0	Function Code=0														
008	CIDBF: idents															
00A	CPAD16: pad															
00C	CADDR32: addr															
010	CSIZE32: size															
014	Packet Checksum															

```
LMEMDUMPMEM (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  16-bit padding pad (fld)
|  32-Bit Address/Offset Bitfield addr (bf)
|  |  Upper 16 Bits of Address/Offset hi (0-15)
|  |  Lower 16 Bits of Address/Offset lo (16-31)
|  32-Bit Word Count Bitfield size (bf)
|  |  Upper 16 Bits of Word Count hi (0-15)
|  |  Lower 16 Bits of Word Count lo (16-31)

typedef struct _MEM_LMEMDUMPMEM_Pkt {

    /* CCSDS header */
```

```
      char                 hdr[8];

      /* Command payload */
      MEM_LMEMDUMPMEM_Prm  pay;

  } MEM_LMEMDUMPMEM_Pkt;
```

Fields:

`CADDR32` **(14.2.0)**

Attribute(s):

Brief: "32-Bit Address/Offset Bitfield"

Instance(s): `addr`

Bitfield containing the full 32 bits of the starting address or offset.

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CPAD16` **(14.3.6)**

Attribute(s):

Brief: "16-bit padding"

Instance(s): `pad`

Explicit padding of 16 bits.

`CSIZE32` **(14.2.3)**

Attribute(s):

Brief: "32-Bit Word Count Bitfield"

Instance(s): `size`

Bitfield containing the full 32-bit word count value.

ITOS Mnemonic: `LMEMDUMPNEXT`

14.1.2 `LMEMDUMPNEXT` (1604/0x644:100)

Context:

Description:

"Send Next Dump Packet" Telecommand Packet

Send the next chunk of memory dump data.

Layout:

Send Next Dump Packet (LMEMDUMPNEXT)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x644:100										
002	SF=3		Sequence Count													
004	Packet Length=5															
006	0	Function Code=100														
008	CIDBF: idents															
00A	Packet Checksum															

```
LMEMDUMPNEXT (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)

typedef struct _MEM_LMEMDUMPNEXT_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPNEXT_Prm pay;

} MEM_LMEMDUMPNEXT_Pkt;
```

Fields:

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

ITOS Mnemonic: `LMEMDUMPPCI`

14.1.3 `LMEMDUMPPCI` (1604/0x644:2)

Context:

Description:

"PCI Device Header Dump" Telecommand Packet

Dump data from a PCI device header.

Layout:

PCI Device Header Dump (LMEMDUMPPCI) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x644:2										
002	SF=3		Sequence Count													
004	Packet Length=7															
006	0	Function Code=2														
008	CIDBF: idents															
00A	CPCIADDRBF: addr															
00C	Packet Checksum															

```
LMEMDUMPPCI (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  PCI Device Header Address Bitfield addr (bf)
|  |  PCI Bus Address bus (0)
|  |  PCI Device Address device (1-5)
|  |  PCI Function Address function (6-7)
|  |  PCI Offset Address offset (8-15)

typedef struct _MEM_LMEMDUMPPCI_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPPCI_Prm  pay;

} MEM_LMEMDUMPPCI_Pkt;
```

Fields:

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CPCIADDRBF` **(14.2.2)**

Attribute(s):

Brief: "PCI Device Header Address Bitfield"

Instance(s): `addr`

Bitfield containing PCI device header address.

ITOS Mnemonic: `LMEMDUMPPOOL`

14.1.4 `LMEMDUMPPOOL` (1604/0x644:7)

Context:

Description:

"Memory Pool Status Dump" Telecommand Packet

Dump the statistics for a memory pool.

Layout:

Memory Pool Status Dump (LMEMDUMPPOOL) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x644:7**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=7**															
006	**0**	**Function Code=7**														
008	**CIDBF:** `idents`															
00A	**CPOOLID:** `poolID`															
00C	**Packet Checksum**															

```
LMEMDUMPPOOL (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  Memory Pool ID poolID (fld)
```

```
typedef struct _MEM_LMEMDUMPPOOL_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPPOOL_Prm pay;

} MEM_LMEMDUMPPOOL_Pkt;
```

Fields:

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CPOOLID` **(14.3.12)**

Attribute(s):

Brief: "Memory Pool ID"

Instance(s): `poolID`

Identifier for the memory pool for which statistics should be dumped.

ITOS Mnemonic: `LMEMDUMPREG`

14.1.5 `LMEMDUMPREG` (1604/0x644:3)

Context:

Description:

"Processor Register Dump" Telecommand Packet

Dump CPU register values.

Layout:

Processor Register Dump (LMEMDUMPREG)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	**Version=0**			**T=1**	**SH**	**APID=0x644:3**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=5**															
006	**0**	**Function Code=3**														
008	**CIDBF:** idents															
00A	**Packet Checksum**															

```
LMEMDUMPREG (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
```

```
typedef struct _MEM_LMEMDUMPREG_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPREG_Prm  pay;

} MEM_LMEMDUMPREG_Pkt;
```

Fields:

CIDBF **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): idents

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

ITOS Mnemonic: LMEMDUMPSYMREL

14.1.6 **LMEMDUMPSYMREL (1604/0x644:9)**

Context:

Description:

"Memory Dump Symbol Relative" Telecommand Packet

Dump memory data starting at an offset relative to a symbol.

Layout:

Memory Dump Symbol Relative (LMEMDUMPSYMREL)
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	**Version=0**			**T=1**	**SH**	**APID=0x644:9**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=55**															
006	**0**	**Function Code=9**														
008	**CIDBF:** idents															
00A	**CPAD16:** pad1															
00C	**CADDR32:** offset															
010	**CSIZE32:** size															
014	**CNAMESIZE:** nameSize								**CPAD8:** pad2							
016	**CPAD16:** pad3															
018	**CNAMECHAR:** name[0-35]															
03C	**Packet Checksum**															

```
LMEMDUMPSYMREL (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  16-bit padding pad1 (fld)
|  32-Bit Address/Offset Bitfield offset (bf)
|  |  Upper 16 Bits of Address/Offset hi (0-15)
|  |  Lower 16 Bits of Address/Offset lo (16-31)
|  32-Bit Word Count Bitfield size (bf)
|  |  Upper 16 Bits of Word Count hi (0-15)
|  |  Lower 16 Bits of Word Count lo (16-31)
|  Symbol Name Length nameSize (fld)
|  8-bit padding pad2 (fld)
|  16-bit padding pad3 (fld)
|  Symbol Name Character name[0-35] (fld)
```

```
typedef struct _MEM_LMEMDUMPSYMREL_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPSYMREL_Prm pay;

} MEM_LMEMDUMPSYMREL_Pkt;
```

Fields:

CADDR32 **(14.2.0)**

Attribute(s):

Brief: "32-Bit Address/Offset Bitfield"

Instance(s): `offset`

Bitfield containing the full 32 bits of the starting address or offset.

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CNAMECHAR` **(14.3.4)**

Attribute(s):

Brief: "Symbol Name Character"

Instance(s): `name`

Character within a symbol name.

`CNAMESIZE` **(14.3.5)**

Attribute(s):

Brief: "Symbol Name Length"

Instance(s): `nameSize`

Length of symbol name, in bytes.

`CPAD16` **(14.3.6)**

Attribute(s):

Brief: "16-bit padding"

Instance(s): `pad1`, `pad3`

Explicit padding of 16 bits.

`CPAD8` **(14.3.7)**

Attribute(s):

Brief: "8-bit padding"

Instance(s): `pad2`

Explicit padding of 8 bits.

`CSIZE32` **(14.2.3)**

Attribute(s):

Brief: "32-Bit Word Count Bitfield"

Instance(s): `size`

Bitfield containing the full 32-bit word count value.

ITOS Mnemonic: `LMEMDUMPSYMVAL`

14.1.7 `LMEMDUMPSYMVAL` (1604/0x644:8)

Context:

Description:

"Memory Symbol Lookup" Telecommand Packet

Dump the value of a symbol.

Layout:

Memory Symbol Lookup (`LMEMDUMPSYMVAL`) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x644:8**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=55**															
006	**0**	**Function Code=8**														
008	**CIDBF:** `idents`															
00A	**CNAMESIZE:** `nameSize`								**CPAD8:** `pad`							
00C	**CNAMECHAR:** `name[0-47]`															
03C	**Packet Checksum**															

```
LMEMDUMPSYMVAL (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
```

```
|  |  Transaction ID tranID (4-15)
|  Symbol Name Length nameSize (fld)
|  8-bit padding pad (fld)
|  Symbol Name Character name[0-47] (fld)

typedef struct _MEM_LMEMDUMPSYMVAL_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMDUMPSYMVAL_Prm pay;

} MEM_LMEMDUMPSYMVAL_Pkt;
```

Fields:

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CNAMECHAR` **(14.3.4)**

Attribute(s):

Brief: "Symbol Name Character"

Instance(s): `name`

Character within a symbol name.

`CNAMESIZE` **(14.3.5)**

Attribute(s):

Brief: "Symbol Name Length"

Instance(s): `nameSize`

Length of symbol name, in bytes.

`CPAD8` **(14.3.7)**

Attribute(s):

Brief: "8-bit padding"

Instance(s): `pad`

Explicit padding of 8 bits.

ITOS Mnemonic: `LMEMLOADMEM`

14.1.8 `LMEMLOADMEM` (1604/0x644:4)

Context:

Description:

"Memory Write" Telecommand Packet

Load data to a region in memory.

Layout:

Memory Write (LMEMLOADMEM) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x644:4**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=55**															
006	**0**	**Function Code=4**														
008	**CIDBF:** `idents`															
00A	**CSIZE16:** `wordCount`															
00C	**CADDR32:** `addr`															
010	**CDATA16:** `data[0-21]`															
03C	**Packet Checksum**															

```
LMEMLOADMEM (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  16-Bit Word count wordCount (fld)
|  32-Bit Address/Offset Bitfield addr (bf)
|  |  Upper 16 Bits of Address/Offset hi (0-15)
|  |  Lower 16 Bits of Address/Offset lo (16-31)
|  16-Bit Data Value data[0-21] (fld)

typedef struct _MEM_LMEMLOADMEM_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
```

```
    MEM_LMEMLOADMEM_Prm  pay;

} MEM_LMEMLOADMEM_Pkt;
```

Fields:

`CADDR32` **(14.2.0)**

Attribute(s):

Brief: "32-Bit Address/Offset Bitfield"

Instance(s): `addr`

Bitfield containing the full 32 bits of the starting address or offset.

`CDATA16` **(14.3.2)**

Attribute(s):

Brief: "16-Bit Data Value"

Instance(s): `data`

16-bit data value.

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CSIZE16` **(14.3.13)**

Attribute(s):

Brief: "16-Bit Word count"

Instance(s): `wordCount`

Number of 32-bit words to load/dump.

ITOS Mnemonic: `LMEMLOADPCI`

14.1.9 `LMEMLOADPCI` (1604/0x644:5)

Context:

Description:

"PCI Device Header Write" Telecommand Packet

Load data to a PCI device header.

Layout:

PCI Device Header Write (LMEMLOADPCI) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x644:5										
002	SF=3		Sequence Count													
004	Packet Length=9															
006	0	Function Code=5														
008	CIDBF: idents															
00A	CPCIADDRBF: addr															
00C	CDATA16: data															
00E	Packet Checksum															

```
LMEMLOADPCI (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  PCI Device Header Address Bitfield addr (bf)
|  |  PCI Bus Address bus (0)
|  |  PCI Device Address device (1-5)
|  |  PCI Function Address function (6-7)
|  |  PCI Offset Address offset (8-15)
|  16-Bit Data Value data (fld)
```

```
typedef struct _MEM_LMEMLOADPCI_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMLOADPCI_Prm  pay;

} MEM_LMEMLOADPCI_Pkt;
```

Fields:

`CDATA16` **(14.3.2)**

Attribute(s):

Brief: "16-Bit Data Value"

Instance(s): `data`

16-bit data value.

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CPCIADDRBF` **(14.2.2)**

Attribute(s):

Brief: "PCI Device Header Address Bitfield"

Instance(s): `addr`

Bitfield containing PCI device header address.

ITOS Mnemonic: `LMEMLOADREG`

14.1.10 `LMEMLOADREG` (1604/0x644:6)

Context:

Description:

"Processor Register Write" Telecommand Packet

Load data to CPU registers.

Layout:

Processor Register Write `(LMEMLOADREG)`
Telecommand Packet Format

0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x644:6										
002	SF=3		Sequence Count													

Processor Register Write (LMEMLOADREG) **Telecommand Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
004	**Packet Length=55**															
006	**0**	**Function Code=6**														
008	**CIDBF:** `idents`															
00A	**CSIZE16:** `wordCount`															
00C	**CADDR32:** `offset`															
010	**CDATA16:** `data[0-21]`															
03C	**Packet Checksum**															

```
LMEMLOADREG (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Target LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  16-Bit Word count wordCount (fld)
|  32-Bit Address/Offset Bitfield offset (bf)
|  |  Upper 16 Bits of Address/Offset hi (0-15)
|  |  Lower 16 Bits of Address/Offset lo (16-31)
|  16-Bit Data Value data[0-21] (fld)
```

```
typedef struct _MEM_LMEMLOADREG_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    MEM_LMEMLOADREG_Prm  pay;

} MEM_LMEMLOADREG_Pkt;
```

Fields:

`CADDR32` **(14.2.0)**

Attribute(s):

Brief: "32-Bit Address/Offset Bitfield"

Instance(s): `offset`

Bitfield containing the full 32 bits of the starting address or offset.

`CDATA16` **(14.3.2)**

Attribute(s):

Brief: "16-Bit Data Value"

Instance(s): `data`

16-bit data value.

`CIDBF` **(14.2.1)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

`CSIZE16` **(14.3.13)**

Attribute(s):

Brief: "16-Bit Word count"

Instance(s): `wordCount`

Number of 32-bit words to load/dump.

14.2 Command Bitfields

14.2.0 CADDR32 (32-Bit Address/Offset Bitfield) Telecommand Bitfield

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Bitfield containing the full 32 bits of the starting address or offset.

Start Stop Size Item Type and Name

```
CADDR32 (bf)
|  Upper 16 Bits of Address/Offset hi (0–15)
|  Lower 16 Bits of Address/Offset lo (16–31)
```

```
typedef struct _MEM_CADDR32_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Upper 16 Bits of Address/Offset */
    unsigned int         hi:16;

    /* Lower 16 Bits of Address/Offset */
    unsigned int         lo:16;

    #elif ENDIANNESS_IS_LITTLE

    /* Lower 16 Bits of Address/Offset */
    unsigned int         lo:16;

    /* Upper 16 Bits of Address/Offset */
    unsigned int         hi:16;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) MEM_CADDR32_Bfl;

typedef union _MEM_CADDR32 {

    /* underlying unsigned int */
    unsigned int                    ui;

    /* struct mapped bitfield */
    struct _MEM_CADDR32_Bfl         bf;

} MEM_CADDR32;
```

Used by:

LMEMDUMPMEM (14.1.1)

14.2.1 **CIDBF** (LAT Unit and Transaction ID Bitfield) Telecommand Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield to contain the LAT unit that is the target of a MEM telecommand and the transaction ID for the MEM telecommand.

Start Stop Size Item Type and Name

```
CIDBF (bf)
|  Target LAT Unit latUnit (0-3)
|  Transaction ID tranID (4-15)

typedef struct _MEM_CIDBF_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Target LAT Unit */
    unsigned short       latUnit:4;

    /* Transaction ID */
    unsigned short       tranID:12;

    #elif ENDIANNESS_IS_LITTLE

    /* Transaction ID */
    unsigned short       tranID:12;

    /* Target LAT Unit */
    unsigned short       latUnit:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) MEM_CIDBF_Bfl;

typedef union _MEM_CIDBF {

    /* underlying unsigned short */
    unsigned short                   us;

    /* struct mapped bitfield */
    struct _MEM_CIDBF_Bfl            bf;
```

```
} MEM_CIDBF;
```

Used by:

```
LMEMDUMPCANCEL (14.1.0)
```

14.2.2 **CPCIADDRBF** (PCI Device Header Address Bitfield) Telecommand Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield containing PCI device header address.

Start Stop Size Item Type and Name

```
CPCIADDRBF (bf)
|  PCI Bus Address bus (0)
|  PCI Device Address device (1-5)
|  PCI Function Address function (6-7)
|  PCI Offset Address offset (8-15)

typedef struct _MEM_CPCIADDRBF_Bfl {

    #if ENDIANNESS_IS_BIG

    /* PCI Bus Address */
    unsigned short       bus:1;

    /* PCI Device Address */
    unsigned short       device:5;

    /* PCI Function Address */
    unsigned short       function:2;

    /* PCI Offset Address */
    unsigned short       offset:8;

    #elif ENDIANNESS_IS_LITTLE

    /* PCI Offset Address */
    unsigned short       offset:8;

    /* PCI Function Address */
    unsigned short       function:2;

    /* PCI Device Address */
    unsigned short       device:5;

    /* PCI Bus Address */
    unsigned short       bus:1;
```

```
    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) MEM_CPCIADDRBF_Bfl;

typedef union _MEM_CPCIADDRBF {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _MEM_CPCIADDRBF_Bfl      bf;

} MEM_CPCIADDRBF;
```

Used by:

`LMEMDUMPPCI (14.1.3)`

14.2.3 `CSIZE32` (32-Bit Word Count Bitfield) Telecommand Bitfield

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Bitfield containing the full 32-bit word count value.

Start Stop Size Item Type and Name

```
CSIZE32 (bf)
│  Upper 16 Bits of Word Count hi (0–15)
│  Lower 16 Bits of Word Count lo (16–31)

typedef struct _MEM_CSIZE32_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Upper 16 Bits of Word Count */
    unsigned int         hi:16;

    /* Lower 16 Bits of Word Count */
    unsigned int         lo:16;

    #elif ENDIANNESS_IS_LITTLE

    /* Lower 16 Bits of Word Count */
    unsigned int         lo:16;

    /* Upper 16 Bits of Word Count */
```

```
    unsigned int         hi:16;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) MEM_CSIZE32_Bfl;

typedef union _MEM_CSIZE32 {

    /* underlying unsigned int */
    unsigned int                   ui;

    /* struct mapped bitfield */
    struct _MEM_CSIZE32_Bfl        bf;

} MEM_CSIZE32;
```

Used by:

LMEMDUMPMEM (14.1.1)

14.3 Command Fields

14.3.0 `CADDRHI` (Upper 16 Bits of Address/Offset) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits

Description:

Upper 16 bits of the 32-bit starting address or offset.

Used by:

???

14.3.1 `CADDRLO` (Lower 16 Bits of Address/Offset) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits

Description:

Lower 16 bits of the 32-bit starting address or offset.

Used by:

???

14.3.2 `CDATA16` (16-Bit Data Value) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

16-bit data value.

Used by:

`LMEMLOADMEM (14.1.8)`

14.3.3 `CLATUNIT` (Target LAT Unit) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	4 bits

Description:

LAT unit that is the target of the MEM telecommand.

Used by:

???

14.3.4 `CNAMECHAR` (Symbol Name Character) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Character within a symbol name.

Used by:

`LMEMDUMPSYMREL (14.1.6)`

14.3.5 `CNAMESIZE` (Symbol Name Length) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Length of symbol name, in bytes.

Used by:

`LMEMDUMPSYMREL (14.1.6)`

14.3.6 `CPAD16` (16-bit padding) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Explicit padding of 16 bits.

Used by:

`LMEMDUMPMEM (14.1.1)`

14.3.7 `CPAD8` (8-bit padding) Telecommand Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Explicit padding of 8 bits.

Used by:

`LMEMDUMPSYMREL (14.1.6)`

14.3.8 `CPCIBUS` (PCI Bus Address) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	1 bit

Description:

Bus portion of a PCI device header address.

Used by:

???

14.3.9 `CPCIDEVICE` (PCI Device Address) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	5 bits

Description:

Device portion of PCI device header address.

Used by:

???

14.3.10 `CPCIFUNCTION` (PCI Function Address) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	2 bits

Description:

Function portion of PCI device header address.

Used by:

???

14.3.11 `CPCIOFFSET` (PCI Offset Address) Telecommand Field

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 8 bits

Description:

Offset portion of PCI device header address.

Used by:

???

14.3.12 `CPOOLID` (Memory Pool ID) Telecommand Field

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Identifier for the memory pool for which statistics should be dumped.

Used by:

`LMEMDUMPPOOL (14.1.4)`

14.3.13 `CSIZE16` (16-Bit Word count) Telecommand Field

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits (2 bytes)

Description:

Number of 32-bit words to load/dump.

Used by:

`LMEMLOADMEM (14.1.8)`

14.3.14 `CSIZEHI` (Upper 16 Bits of Word Count) Telecommand Field

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 16 bits

Description:

Upper 16 bits of the 32-bit word count.

Used by:

???

14.3.15 `CSIZELO` (Lower 16 Bits of Word Count) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits

Description:

Lower 16 bits of the 32-bit word count.

Used by:

???

14.3.16 `CTRANID` (Transaction ID) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	12 bits

Description:

Transaction identifier for the MEM telecommand. This ID is echoed back in the telemetry that is sent in response to the telecommand. It can be used to match telecommands to telemetry.

Used by:

???

14.4 Telemetry Packets

14.4.0 **LMEMPOOLDATA** (785/0x311)

Context:

Description:

"Memory Pool Statistics Dump" Telemetry Packet

Dump of memory pool statistics.

Layout:

Memory Pool Statistics Dump (LMEMPOOLDATA) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x311										
002	SF		Sequence Count													
004	Packet Length=208															
006	Timestamp															
00E	**TIDBF:** idents															
010	**TPOOLID:** poolID															
012	**TPAD16:** pad															
014	**TPOOLFREEBYTES:** freeBytes															
018	**TPOOLFREEBLOCKS:** freeBlocks															
01C	**TPOOLMAXBLKBYTES:** maxBlockBytes															
020	**TPOOLALLOCBYTES:** allocBytes															
024	**TPOOLALLOCBLOCKS:** allocBlocks															

```
LMEMPOOLDATA (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Source LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  Memory Pool ID poolID (fld)
|  16-Bit Padding pad (fld)
|  Memory Pool Free Bytes freeBytes (fld)
|  Memory Pool Free Blocks freeBlocks (fld)
|  Memory Pool Maximum Block Size maxBlockBytes (fld)
|  Memory Pool Allocated Bytes allocBytes (fld)
|  Memory Pool Allocated Blocks allocBlocks (fld)

typedef struct _MEM_LMEMPOOLDATA_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* LAT Unit and Transaction ID Bitfield */
    MEM_TIDBF            idents;

    /* Memory Pool ID */
```

```
    unsigned short       poolID;

    /* 16-Bit Padding */
    unsigned short       pad;

    /* Memory Pool Free Bytes */
    unsigned int         freeBytes;

    /* Memory Pool Free Blocks */
    unsigned int         freeBlocks;

    /* Memory Pool Maximum Block Size */
    unsigned int         maxBlockBytes;

    /* Memory Pool Allocated Bytes */
    unsigned int         allocBytes;

    /* Memory Pool Allocated Blocks */
    unsigned int         allocBlocks;

} MEM_LMEMPOOLDATA_Tlm;
```

Fields:

`TIDBF` **(14.5.0)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield containing the ID of the LAT unit that sent the telemetry and the transaction identifier of the telecommand that requested the telemetry.

`TPAD16` **(14.6.5)**

Attribute(s):

Brief: "16-Bit Padding"

Instance(s): `pad`

Explicit padding of 16 bits.

`TPOOLALLOCBLOCKS` **(14.6.7)**

Attribute(s):

Brief: "Memory Pool Allocated Blocks"

Instance(s): `allocBlocks`

Number of blocks allocated within the memory pool.

`TPOOLALLOCBYTES` **(14.6.8)**

Attribute(s):

Brief: "Memory Pool Allocated Bytes"

Instance(s): `allocBytes`

Number of bytes allocated within the memory pool.

`TPOOLFREEBLOCKS` **(14.6.9)**

Attribute(s):

Brief: "Memory Pool Free Blocks"

Instance(s): `freeBlocks`

Number of free blocks within the memory pool.

`TPOOLFREEBYTES` **(14.6.10)**

Attribute(s):

Brief: "Memory Pool Free Bytes"

Instance(s): `freeBytes`

Number of free bytes within the memory pool.

`TPOOLID` **(14.6.11)**

Attribute(s):

Brief: "Memory Pool ID"

Instance(s): `poolID`

Identifier for the memory pool for which the statistics were dumped.

`TPOOLMAXBLKBYTES` **(14.6.12)**

Attribute(s):

Brief: "Memory Pool Maximum Block Size"

Instance(s): `maxBlockBytes`

Size, in bytes, of the largest free block within the memory pool.

14.4.1 LMEMSIUDATA (788/0x314)

Context:

Description:

"SIU Memory Dump Data" Telemetry Packet

Memory dump data from the SIU.

Layout:

SIU Memory Dump Data (LMEMSIUDATA) Telemetry Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=0	SH	APID=0x314										
002	SF		Sequence Count													
004	Packet Length=2960															
006	Timestamp															
00E	TIDBF: idents															
010	TDUMPADDR: address															
014	TDUMPSIZE: wordCount															
016	TDUMPCMDFUNC: cmdFuncCode															
018	TDUMPDATA: data[0-89]															

```
LMEMSIUDATA (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Source LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  Starting Address address (fld)
|  Word Count wordCount (fld)
|  Dump Type cmdFuncCode (fld)
|  Dump Data Word data[0-89] (fld)
```

```
typedef struct _MEM_LMEMSIUDATA_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* LAT Unit and Transaction ID Bitfield */
    MEM_TIDBF            idents;

    /* Starting Address */
    unsigned int         address;
```

```
    /* Word Count */
    unsigned short       wordCount;

    /* Dump Type */
    unsigned short       cmdFuncCode;

    /* Dump Data Word */
    unsigned int         data[90];

} MEM_LMEMSIUDATA_Tlm;
```

Fields:

`TDUMPADDR` **(14.6.0)**

Attribute(s):

Brief: "Starting Address"

Instance(s): `address`

Address of first 32-bit word of dumped data.

`TDUMPCMDFUNC` **(14.6.1)**

Attribute(s):

Brief: "Dump Type"

Instance(s): `cmdFuncCode`

Function code of telecommand that initiated the dump, which indicates the type of data contained in the dump.

`TDUMPDATA` **(14.6.2)**

Attribute(s):

Brief: "Dump Data Word"

Instance(s): `data`

Memory dump data word.

`TDUMPSIZE` **(14.6.3)**

Attribute(s):

Brief: "Word Count"

Instance(s): `wordCount`

Number of valid 32-bit words of dumped data.

`TIDBF` **(14.5.0)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield containing the ID of the LAT unit that sent the telemetry and the transaction identifier of the telecommand that requested the telemetry.

14.4.2 `LMEMSYMVAL` (786/0x312)

Context:

Description:

"Symbol Value Dump" Telemetry Packet

Dump of a symbol's 32-bit value.

Layout:

Symbol Value Dump (LMEMSYMVAL) **Telemetry Packet Format**																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x312**										
002	**SF**		**Sequence Count**													
004	**Packet Length=464**															
006	**Timestamp**															
00E	**TIDBF:** `idents`															
010	**TSYMVAL:** `symValue`															
014	**TSYMNAMESIZE:** `nameSize`								**TPAD8:** `pad8`							
016	**TPAD16:** `pad16`															
018	**TSYMNAMECHAR:** `name[0-47]`															

```
LMEMSYMVAL (pkt)
|  LAT Unit and Transaction ID Bitfield idents (bf)
|  |  Source LAT Unit latUnit (0-3)
|  |  Transaction ID tranID (4-15)
|  Symbol Value symValue (fld)
|  Symbol Name Length nameSize (fld)
|  8-Bit Padding pad8 (fld)
```

```
|  16-Bit Padding pad16 (fld)
|  Symbol Name Character name[0-47] (fld)

typedef struct _MEM_LMEMSYMVAL_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* LAT Unit and Transaction ID Bitfield */
    MEM_TIDBF            idents;

    /* Symbol Value */
    unsigned int         symValue;

    /* Symbol Name Length */
    unsigned char        nameSize;

    /* 8-Bit Padding */
    unsigned char        pad8;

    /* 16-Bit Padding */
    unsigned short       pad16;

    /* Symbol Name Character */
    unsigned char        name[48];

} MEM_LMEMSYMVAL_Tlm;
```

Fields:

`TIDBF` **(14.5.0)**

Attribute(s):

Brief: "LAT Unit and Transaction ID Bitfield"

Instance(s): `idents`

Bitfield containing the ID of the LAT unit that sent the telemetry and the transaction identifier of the telecommand that requested the telemetry.

`TPAD16` **(14.6.5)**

Attribute(s):

Brief: "16-Bit Padding"

Instance(s): `pad16`

Explicit padding of 16 bits.

`TPAD8` **(14.6.6)**

Attribute(s):

Brief: "8-Bit Padding"

Instance(s): `pad8`

Explicit padding of 8 bits.

`TSYMNAMECHAR` **(14.6.13)**

Attribute(s):

Brief: "Symbol Name Character"

Instance(s): `name`

Character within a symbol name.

`TSYMNAMESIZE` **(14.6.14)**

Attribute(s):

Brief: "Symbol Name Length"

Instance(s): `nameSize`

Length of symbol name, in bytes.

`TSYMVAL` **(14.6.15)**

Attribute(s):

Brief: "Symbol Value"

Instance(s): `symValue`

Value of symbol.

14.5 Telemetry Bitfields

14.5.0 **TIDBF** (LAT Unit and Transaction ID Bitfield) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield containing the ID of the LAT unit that sent the telemetry and the transaction identifier of the telecommand that requested the telemetry.

Start	Stop	Size	Item Type and Name
0	3	4	`TLATUNIT latUnit`
4	15	12	`TTRANID tranID`

```
TIDBF (bf)
|  Source LAT Unit latUnit (0-3)
|  Transaction ID tranID (4-15)
```

```
typedef struct _MEM_TIDBF_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Source LAT Unit */
    unsigned short       latUnit:4;

    /* Transaction ID */
    unsigned short       tranID:12;

    #elif ENDIANNESS_IS_LITTLE

    /* Transaction ID */
    unsigned short       tranID:12;

    /* Source LAT Unit */
    unsigned short       latUnit:4;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) MEM_TIDBF_Bfl;

typedef union _MEM_TIDBF {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
```

```
    struct _MEM_TIDBF_Bfl           bf;

} MEM_TIDBF;
```

Fields:

`TLATUNIT` **(14.6.4)**

Attribute(s):

Brief: "Source LAT Unit"

Instance(s): `latUnit`

LAT unit that sent the telemetry.

`TTRANID` **(14.6.16)**

Attribute(s):

Brief: "Transaction ID"

Instance(s): `tranID`

Transaction identifier from the telecommand that requested the telemetry.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6 Telemetry Fields

14.6.0 `TDUMPADDR` (Starting Address) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Address of first 32-bit word of dumped data.

Used by:

`LMEMSIUDATA (14.4.1)`

14.6.1 `TDUMPCMDFUNC` (Dump Type) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Function code of telecommand that initiated the dump, which indicates the type of data contained in the dump.

Used by:

`LMEMSIUDATA (14.4.1)`

14.6.2 `TDUMPDATA` (Dump Data Word) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Memory dump data word.

Used by:

`LMEMSIUDATA (14.4.1)`

14.6.3 `TDUMPSIZE` (Word Count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of valid 32-bit words of dumped data.

Used by:

`LMEMSIUDATA (14.4.1)`

14.6.4 `TLATUNIT` (Source LAT Unit) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	4 bits

Description:

LAT unit that sent the telemetry.

Used by:

`TIDBF (14.5.0)`

14.6.5 `TPAD16` (16-Bit Padding) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Explicit padding of 16 bits.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.6 `TPAD8` (8-Bit Padding) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Explicit padding of 8 bits.

Used by:

`LMEMSYMVAL (14.4.2)`

14.6.7 `TPOOLALLOCBLOCKS` (Memory Pool Allocated Blocks) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Number of blocks allocated within the memory pool.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.8 `TPOOLALLOCBYTES` (Memory Pool Allocated Bytes) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Number of bytes allocated within the memory pool.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.9 `TPOOLFREEBLOCKS` (Memory Pool Free Blocks) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Number of free blocks within the memory pool.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.10 `TPOOLFREEBYTES` (Memory Pool Free Bytes) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Number of free bytes within the memory pool.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.11 `TPOOLID` (Memory Pool ID) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Identifier for the memory pool for which the statistics were dumped.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.12 `TPOOLMAXBLKBYTES` (Memory Pool Maximum Block Size) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Size, in bytes, of the largest free block within the memory pool.

Used by:

`LMEMPOOLDATA (14.4.0)`

14.6.13 `TSYMNAMECHAR` (Symbol Name Character) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char
Length:	8 bits (1 byte)

Description:

Character within a symbol name.

Used by:

`LMEMSYMVAL (14.4.2)`

14.6.14 `TSYMNAMESIZE` (Symbol Name Length) Telemetry Field

Definition:

Alignment:	1 byte
C type:	unsigned char

Length: 8 bits (1 byte)

Description:

Length of symbol name, in bytes.

Used by:

`LMEMSYMVAL (14.4.2)`

14.6.15 **`TSYMVAL`** (Symbol Value) Telemetry Field

Definition:

Alignment: 4 bytes
C type: unsigned int
Length: 32 bits (4 bytes)

Description:

Value of symbol.

Used by:

`LMEMSYMVAL (14.4.2)`

14.6.16 **`TTRANID`** (Transaction ID) Telemetry Field

Definition:

Alignment: 2 bytes
C type: unsigned short
Length: 12 bits

Description:

Transaction identifier from the telecommand that requested the telemetry.

Used by:

`TIDBF (14.5.0)`

15 PBC Package

15.0 Overview

The PBC package contains routines that are specific to the Primary Boot Code.

The package supports the following functions:

- RAD750 boot and crate initialization

15.1 Command Packets

ITOS Mnemonic: `LPBCBADCMD`

15.1.0 `LBTBAD` (1600/0x640:4)

Context:

Description:

"Invalid boot command" Telecommand Packet

Boot command with an invalid function code. This is used for testing the PBC.

Layout:

<table>
<tr><th colspan="17">Invalid boot command (LBTBAD)
Telecommand Packet Format</th></tr>
<tr><th>0x</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>000</td><td colspan="3">Version=0</td><td>T=1</td><td>SH</td><td colspan="11">APID=0x640:4</td></tr>
<tr><td>002</td><td colspan="2">SF=3</td><td colspan="14">Sequence Count</td></tr>
<tr><td>004</td><td colspan="16">Packet Length=5</td></tr>
<tr><td>006</td><td>0</td><td colspan="15">Function Code=4</td></tr>
<tr><td>008</td><td colspan="16">LATUNIT: latUnit</td></tr>
<tr><td>00A</td><td colspan="16">Packet Checksum</td></tr>
</table>

```
LBTBAD (pkt)
│  LAT unit latUnit (fld)

typedef struct _PBC_LBTBAD_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    PBC_LBTBAD_Prm       pay;

} PBC_LBTBAD_Pkt;
```

Fields:

`LATUNIT` **(15.2.2)**

Attribute(s):

Brief: "LAT unit"

Instance(s): `latUnit`

Target LAT unit for the command.

ITOS Mnemonic: `LPBCERRDUMP`

15.1.1 `LBTERRDUMP` (1600/0x640:2)

Context:

Description:

"Error code pop" Telecommand Packet

Pop the next error code from the PBC error queue.

Layout:

Error code pop (LBTERRDUMP) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x640:2										
002	SF=3		Sequence Count													
004	Packet Length=5															
006	0	Function Code=2														
008	LATUNIT: latUnit															
00A	Packet Checksum															

```
LBTERRDUMP (pkt)
│  LAT unit latUnit (fld)

typedef struct _PBC_LBTERRDUMP_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    PBC_LBTERRDUMP_Prm   pay;

} PBC_LBTERRDUMP_Pkt;
```

Fields:

`LATUNIT` **(15.2.2)**

Attribute(s):

Brief: "LAT unit"

Instance(s): `latUnit`

Target LAT unit for the command.

ITOS Mnemonic: `LPBCRESET`

15.1.2 **LBTRESET** (1600/0x640:1)

Context:

Description:

"Warm reboot" Telecommand Packet

Initiate a warm reboot of the unit.

Layout:

Warm reboot (LBTRESET) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x640:1**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=11**															
006	**0**	**Function Code=1**														
008	**LATUNIT:** latUnit															
00A	**CPAD16:** pad															
00C	**BOOTFLAGS:** bootFlags															
010	**Packet Checksum**															

```
LBTRESET (pkt)
|  LAT unit latUnit (fld)
|  16-bit Padding pad (fld)
|  Boot flags bootFlags (fld)

typedef struct _PBC_LBTRESET_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    PBC_LBTRESET_Prm     pay;

} PBC_LBTRESET_Pkt;
```

Fields:

`BOOTFLAGS` **(15.2.0)**

Attribute(s):

Brief: "Boot flags"

Instance(s): `bootFlags`

Flags to indicate the actions to be taken by the boot code when restarting.

`CPAD16` **(15.2.1)**

Attribute(s):

Brief: "16-bit Padding"

Instance(s): `pad`

Explicit padding of 16 bits.

`LATUNIT` **(15.2.2)**

Attribute(s):

Brief: "LAT unit"

Instance(s): `latUnit`

Target LAT unit for the command.

ITOS Mnemonic: `LPBCRTOSEXEC`

15.1.3 `LBTRTOSEXEC` (1600/0x640:3)

Context:

Description:

"Boot RTOS" Telecommand Packet

Start the RTOS.

Layout:

Boot RTOS (LBTRTOSEXEC) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=1**	**SH**	**APID=0x640:3**										
002	**SF=3**		**Sequence Count**													
004	**Packet Length=11**															

Boot RTOS (LBTRTOSEXEC) Telecommand Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
006	**0**	**Function Code=3**														
008	**LATUNIT:** latUnit															
00A	**CPAD16:** pad															
00C	**BOOTFLAGS:** sbcFlags															
010	**Packet Checksum**															

```
LBTRTOSEXEC (pkt)
|  LAT unit latUnit (fld)
|  16-bit Padding pad (fld)
|  Boot flags sbcFlags (fld)

typedef struct _PBC_LBTRTOSEXEC_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    PBC_LBTRTOSEXEC_Prm  pay;

} PBC_LBTRTOSEXEC_Pkt;
```

Fields:

BOOTFLAGS **(15.2.0)**

Attribute(s):

Brief: "Boot flags"

Instance(s): sbcFlags

Flags to indicate the actions to be taken by the boot code when restarting.

CPAD16 **(15.2.1)**

Attribute(s):

Brief: "16-bit Padding"

Instance(s): pad

Explicit padding of 16 bits.

LATUNIT **(15.2.2)**

Attribute(s):

Brief: "LAT unit"

Instance(s): `latUnit`

Target LAT unit for the command.

ITOS Mnemonic: `LPBCSTART`

15.1.4 `LBTSTART` (1600/0x640:0)

Context:

Description:

"Boot code no-op" Telecommand Packet

No-op command for the PBC.

Layout:

Boot code no-op (LBTSTART) Telecommand Packet Format																
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	Version=0			T=1	SH	APID=0x640:0										
002	SF=3		Sequence Count													
004	Packet Length=5															
006	0	Function Code=0														
008	LATUNIT: latUnit															
00A	Packet Checksum															

```
LBTSTART (pkt)
|  LAT unit latUnit (fld)

typedef struct _PBC_LBTSTART_Pkt {

    /* CCSDS header */
    char                 hdr[8];

    /* Command payload */
    PBC_LBTSTART_Prm     pay;

} PBC_LBTSTART_Pkt;
```

Fields:

`LATUNIT` **(15.2.2)**

Attribute(s):

Brief: "LAT unit"

Instance(s): `latUnit`

Target LAT unit for the command.

15.2 Command Fields

15.2.0 `BOOTFLAGS` (Boot flags) Telecommand Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Flags to indicate the actions to be taken by the boot code when restarting.

Used by:

`LBTRESET (15.1.2)`

15.2.1 `CPAD16` (16-bit Padding) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Explicit padding of 16 bits.

Used by:

`LBTRESET (15.1.2)`

15.2.2 `LATUNIT` (LAT unit) Telecommand Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Target LAT unit for the command.

Used by:

`LBTBAD (15.1.0)`

15.3 Telemetry Packets

15.3.0 LBTEPU0HKP (609/0x261)

Context:

Description:

"EPU 0 Boot Housekeeping Telemetry" Telemetry Packet

Boot housekeeping telemetry from EPU 0.

Layout:

EPU 0 Boot Housekeeping Telemetry (LBTEPU0HKP) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x261**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**BOOTSWMODE:** swMode															
010	**BOOTTOTALERR:** errCount															
012	**BOOTQERR:** qerrCount															
014	**BOOTERRWORD:** qerrWord															
018	**BOOTPKTCOUNT:** tcRcvCount															
01A	**BOOTPKTACCEPT:** tcAccCount															
01C	**BOOTLASTERR:** errWord															
020	**BOOTCOMMAND:** tcLast															
022	**BOOTSPARE1:** spare1															
024	**BOOTSPARE2:** spare2															
026	**BOOTFILESTATE:** fileState															
028	**BOOTFILEPKT:** filePktCount															
02A	**BOOTSCRUBADDRHI:** addrScrubHi															
02C	**BOOTTYPE:** bootType															
02E	**BOOTDUMPCOUNT:** dumpCount															
030	**BOOTDUMPADDR:** dumpAddr															
034	**BOOTDUMPDATA:** dumpData[0-15]															

```
LBTEPU0HKP (pkt)
|  Current PBC operating mode swMode (fld)
|  Number of errors encountered errCount (fld)
|  Number of queued errors qerrCount (fld)
|  Error code qerrWord (fld)
|  Received packet count tcRcvCount (fld)
|  Accepted packet count tcAccCount (fld)
|  Most recent boot error code errWord (fld)
|  Last boot command bitfield tcLast (bf)
|  |  Last Function Code func (0-4)
```

```
|  |  Last APID apid (5-15)
|  Unused location spare1 (fld) ... spare2 (fld)
|  File upload state fileState (fld)
|  File upload packet count filePktCount (fld)
|  Hi 16 bits of software scrub address addrScrubHi (fld)
|  Boot type bootType (fld)
|  Memory dump word count dumpCount (fld)
|  Memory dump address dumpAddr (fld)
|  Memory dump data dumpData[0-15] (fld)

typedef struct _PBC_LBTEPU0HKP_Tlm {

    /* Telemetry header */
    char                 hdr[14];

    /* Current PBC operating mode */
    unsigned short       swMode;

    /* Number of errors encountered */
    unsigned short       errCount;

    /* Number of queued errors */
    unsigned short       qerrCount;

    /* Error code */
    unsigned int         qerrWord;

    /* Received packet count */
    unsigned short       tcRcvCount;

    /* Accepted packet count */
    unsigned short       tcAccCount;

    /* Most recent boot error code */
    unsigned int         errWord;

    /* Last boot command bitfield */
    PBC_BOOTCOMMAND      tcLast;

    /* Unused location */
    unsigned short       spare1;

    /* Unused location */
    unsigned short       spare2;

    /* File upload state */
    unsigned short       fileState;

    /* File upload packet count */
    unsigned short       filePktCount;

    /* Hi 16 bits of software scrub address */
    unsigned short       addrScrubHi;
```

```
    /* Boot type */
    unsigned short       bootType;

    /* Memory dump word count */
    unsigned short       dumpCount;

    /* Memory dump address */
    unsigned int         dumpAddr;

    /* Memory dump data */
    unsigned int         dumpData[16];

} PBC_LBTEPU0HKP_Tlm;
```

Fields:

`BOOTCOMMAND` **(15.4.0)**

Attribute(s):

Brief: "Last boot command bitfield"

Instance(s): `tcLast`

Bitfield to hold APID and function code of command most recently received by the PBC.

`BOOTDUMPADDR` **(15.5.0)**

Attribute(s):

Brief: "Memory dump address"

Instance(s): `dumpAddr`

Address of first memory dump word.

`BOOTDUMPCOUNT` **(15.5.1)**

Attribute(s):

Brief: "Memory dump word count"

Instance(s): `dumpCount`

Number of valid 32-bit memory dump words.

`BOOTDUMPDATA` **(15.5.2)**

Attribute(s):

Brief: "Memory dump data"

Instance(s): `dumpData`

Memory dump data word.

`BOOTERRWORD` **(15.5.3)**

Attribute(s):

Brief: "Error code"

Instance(s): `qerrWord`

Error code at the head of the boot error queue.

`BOOTFILEPKT` **(15.5.4)**

Attribute(s):

Brief: "File upload packet count"

Instance(s): `filePktCount`

Number of file upload packets received by the PBC.

`BOOTFILESTATE` **(15.5.5)**

Attribute(s):

Brief: "File upload state"

Instance(s): `fileState`

State of the PBC's file upload state machine.

`BOOTLASTERR` **(15.5.7)**

Attribute(s):

Brief: "Most recent boot error code"

Instance(s): `errWord`

Error code for the most recent error encountered by the PBC.

`BOOTPKTACCEPT` **(15.5.9)**

Attribute(s):

Brief: "Accepted packet count"

Instance(s): `tcAccCount`

Number of packets accepted by the PBC.

`BOOTPKTCOUNT` **(15.5.10)**

Attribute(s):

Brief: "Received packet count"

Instance(s): `tcRcvCount`

Number of packets received by the PBC.

`BOOTQERR` **(15.5.11)**

Attribute(s):

Brief: "Number of queued errors"

Instance(s): `qerrCount`

Number of errors in the boot error queue.

`BOOTSCRUBADDRHI` **(15.5.12)**

Attribute(s):

Brief: "Hi 16 bits of software scrub address"

Instance(s): `addrScrubHi`

Upper 16 bits of PBC software scrub address.

`BOOTSPARE1` **(15.5.13)**

Attribute(s):

Brief: "Unused location"

Instance(s): `spare1`

`BOOTSPARE2` **(15.5.14)**

Attribute(s):

Brief: "Unused location"

Instance(s): `spare2`

`BOOTSWMODE` **(15.5.15)**

Attribute(s):

Brief: "Current PBC operating mode"

Instance(s): `swMode`

Current PBC operating mode.

`BOOTTOTALERR` **(15.5.16)**

Attribute(s):

Brief: "Number of errors encountered"

Instance(s): `errCount`

Total number of errors encountered by the PBC.

`BOOTTYPE` **(15.5.17)**

Attribute(s):

Brief: "Boot type"

Instance(s): `bootType`

Type of boot.

15.3.1 `LBTHKP` (512/0x200)

Context:

Description:

"Boot housekeeping telemetry" Telemetry Packet

Boot housekeeping telemetry.

Layout:

Boot housekeeping telemetry (LBTHKP) Telemetry Packet Format																
0x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
000	**Version=0**			**T=0**	**SH**	**APID=0x200**										
002	**SF**		**Sequence Count**													
004	**Packet Length=109**															
006	**Timestamp**															
00E	**BOOTSWMODE:** bootSoftwareMode															
010	**BOOTTOTALERR:** totalErrorCount															
012	**BOOTQERR:** queuedErrorCount															
014	**BOOTERRWORD:** errorWord															
018	**BOOTPKTCOUNT:** tcPktReceiveCount															
01A	**BOOTPKTACCEPT:** tcPktAcceptCount															
01C	**BOOTLASTERR:** latestErrorWord															
020	**BOOTCOMMAND:** lastCommand															
022	**BOOTSPARE1:** spare1															
024	**BOOTSPARE2:** spare2															
026	**BOOTFILESTATE:** fileUploadState															
028	**BOOTFILEPKT:** fileUploadPktCnt															
02A	**BOOTSCRUBADDRHI:** scrubAddrHi															
02C	**BOOTTYPE:** bootType															
02E	**BOOTDUMPCOUNT:** memDumpWordCount															
030	**BOOTDUMPADDR:** memDumpAddress															
034	**BOOTDUMPDATA:** memDumpData[0-15]															

```
LBTHKP (pkt)
|  Current PBC operating mode bootSoftwareMode (fld)
|  Number of errors encountered totalErrorCount (fld)
|  Number of queued errors queuedErrorCount (fld)
|  Error code errorWord (fld)
|  Received packet count tcPktReceiveCount (fld)
|  Accepted packet count tcPktAcceptCount (fld)
|  Most recent boot error code latestErrorWord (fld)
|  Last boot command bitfield lastCommand (bf)
|  |  Last Function Code func (0-4)
|  |  Last APID apid (5-15)
|  Unused location spare1 (fld) ... spare2 (fld)
|  File upload state fileUploadState (fld)
|  File upload packet count fileUploadPktCnt (fld)
|  Hi 16 bits of software scrub address scrubAddrHi (fld)
|  Boot type bootType (fld)
|  Memory dump word count memDumpWordCount (fld)
|  Memory dump address memDumpAddress (fld)
|  Memory dump data memDumpData[0-15] (fld)
```

```
typedef struct _PBC_LBTHKP_Tlm {

    /* Telemetry header */
```

```
    char                hdr[14];

    /* Current PBC operating mode */
    unsigned short      bootSoftwareMode;

    /* Number of errors encountered */
    unsigned short      totalErrorCount;

    /* Number of queued errors */
    unsigned short      queuedErrorCount;

    /* Error code */
    unsigned int        errorWord;

    /* Received packet count */
    unsigned short      tcPktReceiveCount;

    /* Accepted packet count */
    unsigned short      tcPktAcceptCount;

    /* Most recent boot error code */
    unsigned int        latestErrorWord;

    /* Last boot command bitfield */
    PBC_BOOTCOMMAND     lastCommand;

    /* Unused location */
    unsigned short      spare1;

    /* Unused location */
    unsigned short      spare2;

    /* File upload state */
    unsigned short      fileUploadState;

    /* File upload packet count */
    unsigned short      fileUploadPktCnt;

    /* Hi 16 bits of software scrub address */
    unsigned short      scrubAddrHi;

    /* Boot type */
    unsigned short      bootType;

    /* Memory dump word count */
    unsigned short      memDumpWordCount;

    /* Memory dump address */
    unsigned int        memDumpAddress;

    /* Memory dump data */
    unsigned int        memDumpData[16];

} PBC_LBTHKP_Tlm;
```

Fields:

`BOOTCOMMAND` **(15.4.0)**

Attribute(s):

Brief: "Last boot command bitfield"

Instance(s): `lastCommand`

Bitfield to hold APID and function code of command most recently received by the PBC.

`BOOTDUMPADDR` **(15.5.0)**

Attribute(s):

Brief: "Memory dump address"

Instance(s): `memDumpAddress`

Address of first memory dump word.

`BOOTDUMPCOUNT` **(15.5.1)**

Attribute(s):

Brief: "Memory dump word count"

Instance(s): `memDumpWordCount`

Number of valid 32-bit memory dump words.

`BOOTDUMPDATA` **(15.5.2)**

Attribute(s):

Brief: "Memory dump data"

Instance(s): `memDumpData`

Memory dump data word.

`BOOTERRWORD` **(15.5.3)**

Attribute(s):

Brief: "Error code"

Instance(s): `errorWord`

Error code at the head of the boot error queue.

`BOOTFILEPKT` **(15.5.4)**

Attribute(s):

Brief: "File upload packet count"

Instance(s): `fileUploadPktCnt`

Number of file upload packets received by the PBC.

`BOOTFILESTATE` **(15.5.5)**

Attribute(s):

Brief: "File upload state"

Instance(s): `fileUploadState`

State of the PBC's file upload state machine.

`BOOTLASTERR` **(15.5.7)**

Attribute(s):

Brief: "Most recent boot error code"

Instance(s): `latestErrorWord`

Error code for the most recent error encountered by the PBC.

`BOOTPKTACCEPT` **(15.5.9)**

Attribute(s):

Brief: "Accepted packet count"

Instance(s): `tcPktAcceptCount`

Number of packets accepted by the PBC.

`BOOTPKTCOUNT` **(15.5.10)**

Attribute(s):

Brief: "Received packet count"

Instance(s): `tcPktReceiveCount`

Number of packets received by the PBC.

`BOOTQERR` **(15.5.11)**

Attribute(s):

Brief: "Number of queued errors"

Instance(s): `queuedErrorCount`

Number of errors in the boot error queue.

`BOOTSCRUBADDRHI` **(15.5.12)**

Attribute(s):

Brief: "Hi 16 bits of software scrub address"

Instance(s): `scrubAddrHi`

Upper 16 bits of PBC software scrub address.

`BOOTSPARE1` **(15.5.13)**

Attribute(s):

Brief: "Unused location"

Instance(s): `spare1`

`BOOTSPARE2` **(15.5.14)**

Attribute(s):

Brief: "Unused location"

Instance(s): `spare2`

`BOOTSWMODE` **(15.5.15)**

Attribute(s):

Brief: "Current PBC operating mode"

Instance(s): `bootSoftwareMode`

Current PBC operating mode.

`BOOTTOTALERR` **(15.5.16)**

Attribute(s):

Brief: "Number of errors encountered"

Instance(s): `totalErrorCount`

Total number of errors encountered by the PBC.

`BOOTTYPE` **(15.5.17)**

Attribute(s):

Brief: "Boot type"

Instance(s): `bootType`

Type of boot.

15.4 Telemetry Bitfields

15.4.0 BOOTCOMMAND (Last boot command bitfield) Telemetry Bitfield

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Bitfield to hold APID and function code of command most recently received by the PBC.

Start	Stop	Size	Item Type and Name
0	4	5	BOOTLASTFUNC func
5	15	11	BOOTLASTAPID apid

```
BOOTCOMMAND (bf)
|  Last Function Code func (0-4)
|  Last APID apid (5-15)

typedef struct _PBC_BOOTCOMMAND_Bfl {

    #if ENDIANNESS_IS_BIG

    /* Last Function Code */
    unsigned short       func:5;

    /* Last APID */
    unsigned short       apid:11;

    #elif ENDIANNESS_IS_LITTLE

    /* Last APID */
    unsigned short       apid:11;

    /* Last Function Code */
    unsigned short       func:5;

    #else

    #error _FILE_ ENDIANNESS NOT DEFINED

    #endif

} __attribute__((packed)) PBC_BOOTCOMMAND_Bfl;

typedef union _PBC_BOOTCOMMAND {

    /* underlying unsigned short */
    unsigned short                  us;

    /* struct mapped bitfield */
    struct _PBC_BOOTCOMMAND_Bfl     bf;
```

```
} PBC_BOOTCOMMAND;
```

Fields:

`BOOTLASTAPID` **(15.5.6)**

Attribute(s):

Brief: "Last APID"

Instance(s): `apid`

APID of last command received by the PBC.

`BOOTLASTFUNC` **(15.5.8)**

Attribute(s):

Brief: "Last Function Code"

Instance(s): `func`

Function code of the last command received by the PBC.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5 Telemetry Fields

15.5.0 `BOOTDUMPADDR` (Memory dump address) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Address of first memory dump word.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.1 `BOOTDUMPCOUNT` (Memory dump word count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of valid 32-bit memory dump words.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.2 `BOOTDUMPDATA` (Memory dump data) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Memory dump data word.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.3 `BOOTERRWORD` (Error code) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Error code at the head of the boot error queue.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.4 `BOOTFILEPKT` (File upload packet count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of file upload packets received by the PBC.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.5 `BOOTFILESTATE` (File upload state) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

State of the PBC's file upload state machine.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.6 `BOOTLASTAPID` (Last APID) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	11 bits

Description:

APID of last command received by the PBC.

Used by:

`BOOTCOMMAND (15.4.0)`

15.5.7 `BOOTLASTERR` (Most recent boot error code) Telemetry Field

Definition:

Alignment:	4 bytes
C type:	unsigned int
Length:	32 bits (4 bytes)

Description:

Error code for the most recent error encountered by the PBC.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.8 `BOOTLASTFUNC` (Last Function Code) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	5 bits

Description:

Function code of the last command received by the PBC.

Used by:

`BOOTCOMMAND (15.4.0)`

15.5.9 `BOOTPKTACCEPT` (Accepted packet count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of packets accepted by the PBC.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.10 `BOOTPKTCOUNT` (Received packet count) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of packets received by the PBC.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.11 **`BOOTQERR`** (Number of queued errors) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Number of errors in the boot error queue.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.12 **`BOOTSCRUBADDRHI`** (Hi 16 bits of software scrub address) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Upper 16 bits of PBC software scrub address.

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.13 **`BOOTSPARE1`** (Unused location) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

`LBTEPU0HKP (15.3.0)`

15.5.14 **`BOOTSPARE2`** (Unused location) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Used by:

LBTEPU0HKP (15.3.0)

15.5.15 **BOOTSWMODE** (Current PBC operating mode) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Current PBC operating mode.

Used by:

LBTEPU0HKP (15.3.0)

15.5.16 **BOOTTOTALERR** (Number of errors encountered) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Total number of errors encountered by the PBC.

Used by:

LBTEPU0HKP (15.3.0)

15.5.17 **BOOTTYPE** (Boot type) Telemetry Field

Definition:

Alignment:	2 bytes
C type:	unsigned short
Length:	16 bits (2 bytes)

Description:

Type of boot.

Used by:

LBTEPU0HKP (15.3.0)

16 Telecommand Packet Index, by APID

APID	FC	Cmd. Packet (L)	Section	Description (L)
0x640	0	LBTSTART	PBC/cmd/_/P.LBTSTART.shtml:%	Boot code no-op
0x640	1	LBTRESET	PBC/cmd/_/P.LBTRESET.shtml:%	Warm reboot
0x640	2	LBTERRDUMP	PBC/cmd/_/P.LBTERRDUMP.shtml:%	Error code pop
0x640	3	LBTRTOSEXEC	PBC/cmd/_/P.LBTRTOSEXEC.shtml:%	Boot RTOS
0x640	4	LBTBAD	PBC/cmd/_/P.LBTBAD.shtml:%	Invalid boot command
0x641	0	LFILUPLSTART	FILE/cmd/_/P.LFILUPLSTART.shtml:%	File Upload Start
0x641	1	LFILUPLCANCEL	FILE/cmd/_/P.LFILUPLCANCEL.shtml:%	File Upload Cancel
0x641	2	LFILUPLCOMMIT	FILE/cmd/_/P.LFILUPLCOMMIT.shtml:%	File Upload Commit
0x641	3	LFILUPLDATA	FILE/cmd/_/P.LFILUPLDATA.shtml:%	File Upload Data
0x641	4	LFILUPLEPU	FILE/cmd/_/P.LFILUPLEPU.shtml:%	File Upload to EPU
[illegible]	[illegible]	LMEMDUMPMEM	MEM/cmd/_/P.LMEMDUMPMEM.shtml:%	Memory Data Dump
[illegible]	[illegible]	LMEMDUMPCANCEL	[illegible]	Memory Dump Cancel
[illegible]	[illegible]	LMEMDUMPPCI	[illegible]	PCI Device Header Dump
[illegible]	[illegible]	LMEMDUMPREG	[illegible]	Processor Register Dump
[illegible]	[illegible]	LMEMLOADMEM	[illegible]	Memory Write
[illegible]	[illegible]	LMEMLOADPCI	[illegible]	PCI Device Header Write
[illegible]	[illegible]	LMEMLOADREG	[illegible]	Processor Register Write
[illegible]	[illegible]	[illegible]	[illegible]	Memory Pool Status Dump
[illegible]	[illegible]	[illegible]	[illegible]	Memory Symbol Lookup
[illegible]	[illegible]	[illegible]	[illegible]	Memory Dump Symbol Relative
[illegible]	[illegible]	[illegible]	[illegible]	Send Next Dump Packet
[illegible]	[illegible]	Restart	[illegible]	Restart and initialize Thermal Control[illegible]
[illegible]	[illegible]	Start	[illegible]	Start control
[illegible]	[illegible]	Stop	[illegible]	Terminate Thermal Control processing
[illegible]	[illegible]	SetMode	[illegible]	Set thermal control mode to active or[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	Set heater to always on, or off or auto[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	Set control parameters to new values
[illegible]	[illegible]	[illegible]	[illegible]	Set LTO telemetry frequency, 0 is off
[illegible]	[illegible]	[illegible]	[illegible]	File Delete
[illegible]	[illegible]	[illegible]	[illegible]	File Copy Local
[illegible]	[illegible]	[illegible]	[illegible]	Directory Create
[illegible]	[illegible]	[illegible]	[illegible]	Directory Delete
[illegible]	[illegible]	[illegible]	[illegible]	File Dump CTDB
[illegible]	[illegible]	[illegible]	[illegible]	Directory Dump
[illegible]	[illegible]	[illegible]	[illegible]	File System Status
[illegible]	[illegible]	[illegible]	[illegible]	Request a Housekeeping Diagnostic [illegible]
[illegible]	[illegible]	[illegible]	[illegible]	System Reset
[illegible]	[illegible]	[illegible]	[illegible]	Stop Diagnostic Sample
[illegible]	[illegible]	[illegible]	[illegible]	Change task messaging level
[illegible]	[illegible]	[illegible]	[illegible]	Change task command confirmation [illegible]
[illegible]	[illegible]	[illegible]	[illegible]	CAL Low Rate Science Counters
[illegible]	[illegible]	[illegible]	[illegible]	TKR Low Rate Science Counters
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] Tile Counter (Pair)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] Tile Counters (All)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] Active Counter
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] Broadcast Message
[illegible]	[illegible]	[illegible]	[illegible]	SC Ancillary Broadcast Message
[illegible]	[illegible]	[illegible]	[illegible]	SC Timetone Broadcast Message

17 Telecommand Packet Index, by Mnemonic (ITOS)

Cmd. Packet (I)	APID	FC	Section	Description (I)
CmdResponse	0x695	1	LCM/cmd/_/P.cmdResponse.shtml:%	Change task command confirmation le
LFILUPLCANCEL	0x641	1	FILE/cmd/_/P.LFILUPLCANCEL.shtml:%	File Upload Cancel
LFILUPLCOMMIT	0x641	2	FILE/cmd/_/P.LFILUPLCOMMIT.shtml:%	File Upload Commit
LFILUPLDATA	0x641	3	FILE/cmd/_/P.LFILUPLDATA.shtml:%	File Upload Data
LFILUPLEPU	0x641	4	FILE/cmd/_/P.LFILUPLEPU.shtml:%	File Upload to EPU
LFILUPLSTART	0x641	0	FILE/cmd/_/P.LFILUPLSTART.shtml:%	File Upload Start
LHKREQDIAGPKT	0x650	0	LHK/cmd/_/P.ReqDiagPacket.shtml:%	Request a Housekeeping Diagnostic P
LHKSTOPDIAG	0x650	2	LHK/cmd/_/P.StopDiag.shtml:%	Stop Diagnostic Sample
LHKSYSRESET	0x650	1	LHK/cmd/_/P.SysReset.shtml:%	System Reset
LLFSDIRCREATE	0x648	2	LFS/cmd/_/P.LLFSDIRCREATE.shtml:%	Directory Create
LLFSDIRDELETE	0x648	3	LFS/cmd/_/P.LLFSDIRDELETE.shtml:%	Directory Delete
LLFSDIRDUMP	0x648	5	LFS/cmd/_/P.LLFSDIRDUMP.shtml:%	Directory Dump
LLFSFILECOPY	0x648	1	LFS/cmd/_/P.LLFSFILECOPY.shtml:%	File Copy Local
LLFSFILEDELETE	0x648	0	LFS/cmd/_/P.LLFSFILEDELETE.shtml:%	File Delete
LLFSFILEDUMPC	0x648	4	LFS/cmd/_/P.LLFSFILEDUMPC.shtml:%	File Dump CTDB
LLFSSYSSTATUS	0x648	6	LFS/cmd/_/P.LLFSSYSSTATUS.shtml:%	File System Status
LMCACDTILEALL	0x69C	3	LMC/cmd/_/P.acd_tile_all.shtml:%	ACD Tile Counters (All)
LMCACDTILEPAIR	0x69C	2	LMC/cmd/_/P.acd_tile_pair.shtml:%	ACD Tile Counter (Pair)
LMCCALLRS	0x69C	0	LMC/cmd/_/P.cal_lrs.shtml:%	CAL Low Rate Science Counters
LMCSTOPCOUNT	0x69C	4	LMC/cmd/_/P.stop_count.shtml:%	Stop Active Counter
LMCTKRLRS	0x69C	1	LMC/cmd/_/P.tkr_lrs.shtml:%	TKR Low Rate Science Counters
LMEMDUMPCANCEL	0x644	1	MEM/cmd/_/P.LMEMDUMPCANCEL.shtml:%	Memory Dump Cancel
LMEMDUMPMEM	0x644	0	MEM/cmd/_/P.LMEMDUMPMEM.shtml:%	Memory Data Dump
LMEMDUMPNEXT	0x644	100	MEM/cmd/_/P.LMEMDUMPNEXT.shtml:%	Send Next Dump Packet
LMEMDUMPPCI	0x644	2	MEM/cmd/_/P.LMEMDUMPPCI.shtml:%	PCI Device Header Dump
LMEMDUMPPOOL	0x644	7	MEM/cmd/_/P.LMEMDUMPPOOL.shtml:%	Memory Pool Status Dump
LMEMDUMPREG	0x644	3	MEM/cmd/_/P.LMEMDUMPREG.shtml:%	Processor Register Dump
LMEMDUMPSYMREL	0x644	9	MEM/cmd/_/P.LMEMDUMPSYMREL.shtml:%	Memory Dump Symbol Relative
LMEMDUMPSYMVAL	0x644	8	MEM/cmd/_/P.LMEMDUMPSYMVAL.shtml:%	Memory Symbol Lookup
LMEMLOADMEM	0x644	4	MEM/cmd/_/P.LMEMLOADMEM.shtml:%	Memory Write
LMEMLOADPCI	0x644	5	MEM/cmd/_/P.LMEMLOADPCI.shtml:%	PCI Device Header Write
LMEMLOADREG	0x644	6	MEM/cmd/_/P.LMEMLOADREG.shtml:%	Processor Register Write
LPBCBADCMD	0x640	4	PBC/cmd/_/P.LBTBAD.shtml:%	Invalid boot command
LPBCERRDUMP	0x640	2	PBC/cmd/_/P.LBTERRDUMP.shtml:%	Error code pop
LPBCRESET	0x640	1	PBC/cmd/_/P.LBTRESET.shtml:%	Warm reboot
LPBCRTOSEXEC	0x640	3	PBC/cmd/_/P.LBTRTOSEXEC.shtml:%	Boot RTOS
LPBCSTART	0x640	0	PBC/cmd/_/P.LBTSTART.shtml:%	Boot code no-op
LTCHtrOnOffCtl	0x645	5	LTC/cmd/_/P.HtrOnOffCtl.shtml:%	Set heater to always on, or off or autor
LTCReStart	0x645	1	LTC/cmd/_/P.ReStart.shtml:%	Restart and initialize Thermal Control
LTCSetMode	0x645	4	LTC/cmd/_/P.SetMode.shtml:%	Set thermal control mode to active or p
LTCSetParam	0x645	6	LTC/cmd/_/P.SetParam.shtml:%	Set control parameters to new values.
LTCSetTlmFreq	0x645	7	LTC/cmd/_/P.SetTlmFreq.shtml:%	Set LTC telemetry frequency, 0 is off.
LTCStart	0x645	2	LTC/cmd/_/P.Start.shtml:%	Start control
LTCStop	0x645	3	LTC/cmd/_/P.Stop.shtml:%	Terminate Thermal Control processing
MsgResponse	0x695	0	LCM/cmd/_/P.msgResponse.shtml:%	Change task messaging level

18 Telecommand Packet Index, by Name (LCAT)

Cmd. Packet (L)	APID	FC	Section	Description (L)
acd_tile_all	0x69C	3	LMC/cmd/_/P.acd_tile_all.shtml:%	ACD Tile Counters (All)
acd_tile_pair	0x69C	2	LMC/cmd/_/P.acd_tile_pair.shtml:%	ACD Tile Counter (Pair)
cal_lrs	0x69C	0	LMC/cmd/_/P.cal_lrs.shtml:%	CAL Low Rate Science Counters
cmdResponse	0x695	1	LCM/cmd/_/P.cmdResponse.shtml:%	Change task command confirmation le
HtrOnOffCtl	0x645	5	LTC/cmd/_/P.HtrOnOffCtl.shtml:%	Set heater to always on, or off or autor
LBTBAD	0x640	4	PBC/cmd/_/P.LBTBAD.shtml:%	Invalid boot command
LBTERRDUMP	0x640	2	PBC/cmd/_/P.LBTERRDUMP.shtml:%	Error code pop
LBTRESET	0x640	1	PBC/cmd/_/P.LBTRESET.shtml:%	Warm reboot
LBTRTOSEXEC	0x640	3	PBC/cmd/_/P.LBTRTOSEXEC.shtml:%	Boot RTOS
LBTSTART	0x640	0	PBC/cmd/_/P.LBTSTART.shtml:%	Boot code no-op
LFILUPLCANCEL	0x641	1	FILE/cmd/_/P.LFILUPLCANCEL.shtml:%	File Upload Cancel
LFILUPLCOMMIT	0x641	2	FILE/cmd/_/P.LFILUPLCOMMIT.shtml:%	File Upload Commit
LFILUPLDATA	0x641	3	FILE/cmd/_/P.LFILUPLDATA.shtml:%	File Upload Data
LFILUPLEPU	0x641	4	FILE/cmd/_/P.LFILUPLEPU.shtml:%	File Upload to EPU
LFILUPLSTART	0x641	0	FILE/cmd/_/P.LFILUPLSTART.shtml:%	File Upload Start
LLFSDIRCREATE	0x648	2	LFS/cmd/_/P.LLFSDIRCREATE.shtml:%	Directory Create
LLFSDIRDELETE	0x648	3	LFS/cmd/_/P.LLFSDIRDELETE.shtml:%	Directory Delete
LLFSDIRDUMP	0x648	5	LFS/cmd/_/P.LLFSDIRDUMP.shtml:%	Directory Dump
LLFSFILECOPY	0x648	1	LFS/cmd/_/P.LLFSFILECOPY.shtml:%	File Copy Local
LLFSFILEDELETE	0x648	0	LFS/cmd/_/P.LLFSFILEDELETE.shtml:%	File Delete
LLFSFILEDUMPC	0x648	4	LFS/cmd/_/P.LLFSFILEDUMPC.shtml:%	File Dump CTDB
LLFSSYSSTATUS	0x648	6	LFS/cmd/_/P.LLFSSYSSTATUS.shtml:%	File System Status
LLSMSIANCILLARY	0x701	2	LSM/cmd/_/P.LLSMSIANCILLARY.shtml:%	SC Ancillary Broadcast Message
LLSMSIATTITUDE	0x701	1	LSM/cmd/_/P.LLSMSIATTITUDE.shtml:%	SC Attitude Broadcast Message
LLSMSITIMETONE	0x701	3	LSM/cmd/_/P.LLSMSITIMETONE.shtml:%	SC Timetone Broadcast Message
LMEMDUMPCANCEL	0x644	1	MEM/cmd/_/P.LMEMDUMPCANCEL.shtml:%	Memory Dump Cancel
LMEMDUMPMEM	0x644	0	MEM/cmd/_/P.LMEMDUMPMEM.shtml:%	Memory Data Dump
LMEMDUMPNEXT	0x644	100	MEM/cmd/_/P.LMEMDUMPNEXT.shtml:%	Send Next Dump Packet
LMEMDUMPPCI	0x644	2	MEM/cmd/_/P.LMEMDUMPPCI.shtml:%	PCI Device Header Dump
LMEMDUMPPOOL	0x644	7	MEM/cmd/_/P.LMEMDUMPPOOL.shtml:%	Memory Pool Status Dump
LMEMDUMPREG	0x644	3	MEM/cmd/_/P.LMEMDUMPREG.shtml:%	Processor Register Dump
LMEMDUMPSYMREL	0x644	9	MEM/cmd/_/P.LMEMDUMPSYMREL.shtml:%	Memory Dump Symbol Relative
LMEMDUMPSYMVAL	0x644	8	MEM/cmd/_/P.LMEMDUMPSYMVAL.shtml:%	Memory Symbol Lookup
LMEMLOADMEM	0x644	4	MEM/cmd/_/P.LMEMLOADMEM.shtml:%	Memory Write
LMEMLOADPCI	0x644	5	MEM/cmd/_/P.LMEMLOADPCI.shtml:%	PCI Device Header Write
LMEMLOADREG	0x644	6	MEM/cmd/_/P.LMEMLOADREG.shtml:%	Processor Register Write
msgResponse	0x695	0	LCM/cmd/_/P.msgResponse.shtml:%	Change task messaging level
ReqDiagPacket	0x650	0	LHK/cmd/_/P.ReqDiagPacket.shtml:%	Request a Housekeeping Diagnostic P
ReStart	0x645	1	LTC/cmd/_/P.ReStart.shtml:%	Restart and initialize Thermal Control
SetMode	0x645	4	LTC/cmd/_/P.SetMode.shtml:%	Set thermal control mode to active or p
SetParam	0x645	6	LTC/cmd/_/P.SetParam.shtml:%	Set control parameters to new values.
SetTlmFreq	0x645	7	LTC/cmd/_/P.SetTlmFreq.shtml:%	Set LTC telemetry frequency, 0 is off.
Start	0x645	2	LTC/cmd/_/P.Start.shtml:%	Start control
Stop	0x645	3	LTC/cmd/_/P.Stop.shtml:%	Terminate Thermal Control processing
stop_count	0x69C	4	LMC/cmd/_/P.stop_count.shtml:%	Stop Active Counter
StopDiag	0x650	2	LHK/cmd/_/P.StopDiag.shtml:%	Stop Diagnostic Sample
SysReset	0x650	1	LHK/cmd/_/P.SysReset.shtml:%	System Reset
tkr_lrs	0x69C	1	LMC/cmd/_/P.tkr_lrs.shtml:%	TKR Low Rate Science Counters

19 Telecommand Struct Index, by Name (LCAT)

Struct (L)	Cmd. Packet (L)	Section	Description (L)

20 Telecommand Bitfield Index, by Name (LCAT)

Bitfield (L)	Cmd. Packet (L)	Section	Description (L)
CADDR32	LMEMDUMPMEM	MEM/cmd/_/B.CADDR32.shtml:%	32-Bit Address/Offset Bitfield
CADDR32	LMEMDUMPSYMREL	MEM/cmd/_/B.CADDR32.shtml:%	32-Bit Address/Offset Bitfield
CADDR32	LMEMLOADMEM	MEM/cmd/_/B.CADDR32.shtml:%	32-Bit Address/Offset Bitfield
CADDR32	LMEMLOADREG	MEM/cmd/_/B.CADDR32.shtml:%	32-Bit Address/Offset Bitfield
CIDBF	LMEMDUMPCANCEL	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPMEM	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPNEXT	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPPCI	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPPOOL	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPREG	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPSYMREL	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMDUMPSYMVAL	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMLOADMEM	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMLOADPCI	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CIDBF	LMEMLOADREG	MEM/cmd/_/B.CIDBF.shtml:%	LAT Unit and Transaction ID Bitfield
CPCIADDRBF	LMEMDUMPPCI	MEM/cmd/_/B.CPCIADDRBF.shtml:%	PCI Device Header Address Bitfield
CPCIADDRBF	LMEMLOADPCI	MEM/cmd/_/B.CPCIADDRBF.shtml:%	PCI Device Header Address Bitfield
CSIZE32	LMEMDUMPMEM	MEM/cmd/_/B.CSIZE32.shtml:%	32-Bit Word Count Bitfield
CSIZE32	LMEMDUMPSYMREL	MEM/cmd/_/B.CSIZE32.shtml:%	32-Bit Word Count Bitfield

21 Telecommand Field Index, by Name (LCAT)

Field (L)	Cmd. Packet (L)	Section	Description (L)
APID	ReqDiagPacket	LHK/cmd/_/F.APID.shtml:%	Application ID parameter
BOOTFLAGS	LBTRESET	PBC/cmd/_/F.BOOTFLAGS.shtml:%	Boot flags
BOOTFLAGS	LBTRTOSEXEC	PBC/cmd/_/F.BOOTFLAGS.shtml:%	Boot flags
CADDRHI	LMEMDUMPMEM	MEM/cmd/_/F.CADDRHI.shtml:%	Upper 16 Bits of Address/Offset
CADDRHI	LMEMDUMPSYMREL	MEM/cmd/_/F.CADDRHI.shtml:%	Upper 16 Bits of Address/Offset
CADDRHI	LMEMLOADMEM	MEM/cmd/_/F.CADDRHI.shtml:%	Upper 16 Bits of Address/Offset
CADDRHI	LMEMLOADREG	MEM/cmd/_/F.CADDRHI.shtml:%	Upper 16 Bits of Address/Offset
CADDRLO	LMEMDUMPMEM	MEM/cmd/_/F.CADDRLO.shtml:%	Lower 16 Bits of Address/Offset
CADDRLO	LMEMDUMPSYMREL	MEM/cmd/_/F.CADDRLO.shtml:%	Lower 16 Bits of Address/Offset
CADDRLO	LMEMLOADMEM	MEM/cmd/_/F.CADDRLO.shtml:%	Lower 16 Bits of Address/Offset
CADDRLO	LMEMLOADREG	MEM/cmd/_/F.CADDRLO.shtml:%	Lower 16 Bits of Address/Offset
cal_mask	cal_lrs	LMC/cmd/_/F.cal_mask.shtml:%	CAL LRS Mask
CDATA16	LMEMLOADMEM	MEM/cmd/_/F.CDATA16.shtml:%	16-Bit Data Value
CDATA16	LMEMLOADPCI	MEM/cmd/_/F.CDATA16.shtml:%	16-Bit Data Value
CDATA16	LMEMLOADREG	MEM/cmd/_/F.CDATA16.shtml:%	16-Bit Data Value
CLATUNIT	LMEMDUMPCANCEL	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPMEM	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPNEXT	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPPCI	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPPOOL	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPREG	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPSYMREL	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMDUMPSYMVAL	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMLOADMEM	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMLOADPCI	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CLATUNIT	LMEMLOADREG	MEM/cmd/_/F.CLATUNIT.shtml:%	Target LAT Unit
CNAMECHAR	LMEMDUMPSYMREL	MEM/cmd/_/F.CNAMECHAR.shtml:%	Symbol Name Character
CNAMECHAR	LMEMDUMPSYMVAL	MEM/cmd/_/F.CNAMECHAR.shtml:%	Symbol Name Character
CNAMESIZE	LMEMDUMPSYMREL	MEM/cmd/_/F.CNAMESIZE.shtml:%	Symbol Name Length
CNAMESIZE	LMEMDUMPSYMVAL	MEM/cmd/_/F.CNAMESIZE.shtml:%	Symbol Name Length
count	acd_tile_all	LMC/cmd/_/F.count.shtml:%	Sample count
count	acd_tile_pair	LMC/cmd/_/F.count.shtml:%	Sample count
count	cal_lrs	LMC/cmd/_/F.count.shtml:%	Sample count
count	tkr_lrs	LMC/cmd/_/F.count.shtml:%	Sample count
counter_opcode	stop_count	LMC/cmd/_/F.counter_opcode.shtml:%	Counter type opcode
CPAD16	LBTRESET	PBC/cmd/_/F.CPAD16.shtml:%	16-bit Padding
CPAD16	LBTRTOSEXEC	PBC/cmd/_/F.CPAD16.shtml:%	16-bit Padding
CPAD16	LMEMDUMPMEM	MEM/cmd/_/F.CPAD16.shtml:%	16-bit padding
CPAD16	LMEMDUMPSYMREL	MEM/cmd/_/F.CPAD16.shtml:%	16-bit padding
CPAD8	LMEMDUMPSYMREL	MEM/cmd/_/F.CPAD8.shtml:%	8-bit padding
CPAD8	LMEMDUMPSYMVAL	MEM/cmd/_/F.CPAD8.shtml:%	8-bit padding
CPCIBUS	LMEMDUMPPCI	MEM/cmd/_/F.CPCIBUS.shtml:%	PCI Bus Address
CPCIBUS	LMEMLOADPCI	MEM/cmd/_/F.CPCIBUS.shtml:%	PCI Bus Address
CPCIDEVICE	LMEMDUMPPCI	MEM/cmd/_/F.CPCIDEVICE.shtml:%	PCI Device Address
CPCIDEVICE	LMEMLOADPCI	MEM/cmd/_/F.CPCIDEVICE.shtml:%	PCI Device Address
CPCIFUNCTION	LMEMDUMPPCI	MEM/cmd/_/F.CPCIFUNCTION.shtml:%	PCI Function Address
CPCIFUNCTION	LMEMLOADPCI	MEM/cmd/_/F.CPCIFUNCTION.shtml:%	PCI Function Address
CPCIOFFSET	LMEMDUMPPCI	MEM/cmd/_/F.CPCIOFFSET.shtml:%	PCI Offset Address
CPCIOFFSET	LMEMLOADPCI	MEM/cmd/_/F.CPCIOFFSET.shtml:%	PCI Offset Address
CPOOLID	LMEMDUMPPOOL	MEM/cmd/_/F.CPOOLID.shtml:%	Memory Pool ID
CSIZE16	LMEMLOADMEM	MEM/cmd/_/F.CSIZE16.shtml:%	16-Bit Word count

Field (L)	Cmd. Packet (L)	Section	Description (L)
CSIZE16	LMEMLOADREG	MEM/cmd/_/F.CSIZE16.shtml:%	16-Bit Word count
CSIZEHI	LMEMDUMPMEM	MEM/cmd/_/F.CSIZEHI.shtml:%	Upper 16 Bits of Word Count
CSIZEHI	LMEMDUMPSYMREL	MEM/cmd/_/F.CSIZEHI.shtml:%	Upper 16 Bits of Word Count
CSIZELO	LMEMDUMPMEM	MEM/cmd/_/F.CSIZELO.shtml:%	Lower 16 Bits of Word Count
CSIZELO	LMEMDUMPSYMREL	MEM/cmd/_/F.CSIZELO.shtml:%	Lower 16 Bits of Word Count
CTRANID	LMEMDUMPCANCEL	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPMEM	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPNEXT	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPPCI	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPPOOL	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPREG	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPSYMREL	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMDUMPSYMVAL	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMLOADMEM	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMLOADPCI	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
CTRANID	LMEMLOADREG	MEM/cmd/_/F.CTRANID.shtml:%	Transaction ID
DiagInterval	ReqDiagPacket	LHK/cmd/_/F.DiagInterval.shtml:%	Diagnostic sample interval
DiagPktCnt	ReqDiagPacket	LHK/cmd/_/F.DiagPktCnt.shtml:%	Diagnostic sample Packet Count
FILEDATA	LFILUPLDATA	FILE/cmd/_/F.FILEDATA.shtml:%	File Data
FILEFLAGS	LFILUPLCOMMIT	FILE/cmd/_/F.FILEFLAGS.shtml:%	File Commit Flags
FILEID	LFILUPLCOMMIT	FILE/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LFILUPLEPU	FILE/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSDIRCREATE	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSDIRDELETE	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSDIRDUMP	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSFILECOPY	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSFILEDELETE	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSFILEDUMPC	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FILEID	LLFSSYSSTATUS	LFS/cmd/_/F.FILEID.shtml:%	File Storage ID
FileId	ReStart	LTC/cmd/_/F.FileId.shtml:%	File identification number
FileId	SysReset	LHK/cmd/_/F.FileId.shtml:%	Configuration File ID
FILEOFFSET	LFILUPLDATA	FILE/cmd/_/F.FILEOFFSET.shtml:%	File Data Offset
FILESIZE	LFILUPLSTART	FILE/cmd/_/F.FILESIZE.shtml:%	File Size
FLOAT	SetParam	LTC/cmd/_/F.FLOAT.shtml:%	32-bit real number.
interval	acd_tile_all	LMC/cmd/_/F.interval.shtml:%	Sample interval in milliseconds
interval	acd_tile_pair	LMC/cmd/_/F.interval.shtml:%	Sample interval in milliseconds
interval	cal_lrs	LMC/cmd/_/F.interval.shtml:%	Sample interval in milliseconds
interval	tkr_lrs	LMC/cmd/_/F.interval.shtml:%	Sample interval in milliseconds
ITC_CmdAction	cmdResponse	LCM/cmd/_/F.ITC_CmdAction.shtml:%	Actions task can take (forward or exe
ITC_CmdClass	cmdResponse	LCM/cmd/_/F.ITC_CmdClass.shtml:%	Spacecraft command class (normal o
ITC_CmdLevel	cmdResponse	LCM/cmd/_/F.ITC_CmdLevel.shtml:%	Response level
ITC_NodeId	cmdResponse	LCM/cmd/_/F.ITC_NodeId.shtml:%	ITC node ID
ITC_NodeId	msgResponse	LCM/cmd/_/F.ITC_NodeId.shtml:%	ITC node ID
ITC_TaskId	cmdResponse	LCM/cmd/_/F.ITC_TaskId.shtml:%	ITC task ID
ITC_TaskId	msgResponse	LCM/cmd/_/F.ITC_TaskId.shtml:%	ITC task ID
LATUNIT	LBTBAD	PBC/cmd/_/F.LATUNIT.shtml:%	LAT unit
LATUNIT	LBTERRDUMP	PBC/cmd/_/F.LATUNIT.shtml:%	LAT unit
LATUNIT	LBTRESET	PBC/cmd/_/F.LATUNIT.shtml:%	LAT unit
LATUNIT	LBTRTOSEXEC	PBC/cmd/_/F.LATUNIT.shtml:%	LAT unit
LATUNIT	LBTSTART	PBC/cmd/_/F.LATUNIT.shtml:%	LAT unit
LATUNIT	LFILUPLCANCEL	FILE/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID
LATUNIT	LFILUPLCOMMIT	FILE/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID

Field (L)	Cmd. Packet (L)	Section	Description (L)
LATUNIT	LFILUPLEPU	FILE/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID
LATUNIT	LLFSDIRCREATE	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSDIRDELETE	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSDIRDUMP	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSFILECOPY	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSFILEDELETE	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSFILEDUMPC	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LATUNIT	LLFSSYSSTATUS	LFS/cmd/_/F.LATUNIT.shtml:%	LAT Unit ID and Transaction ID
LLSMANCFLAGS	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMANCFLAGS.shtml:%	Ancillary SC flags
LLSMANCMODE	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMANCMODE.shtml:%	Ancillary GNC mode
LLSMANCPOS	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMANCPOS.shtml:%	Ancillary position element
LLSMANCSSR	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMANCSSR.shtml:%	Ancillary SSR usage
LLSMANCVEL	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMANCVEL.shtml:%	Ancillary velocity element
LLSMATTAVEC	LLSMSIATTITUDE	LSM/cmd/_/F.LLSMATTAVEC.shtml:%	Attitude angular velocity element
LLSMATTQUAT	LLSMSIATTITUDE	LSM/cmd/_/F.LLSMATTQUAT.shtml:%	Attitude quaternion element
LLSMTIMEFLAGS	LLSMSITIMETONE	LSM/cmd/_/F.LLSMTIMEFLAGS.shtml:%	Timetone SC flags
LLSMTIMESEC	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMTIMESEC.shtml:%	Timestamp seconds
LLSMTIMESEC	LLSMSIATTITUDE	LSM/cmd/_/F.LLSMTIMESEC.shtml:%	Timestamp seconds
LLSMTIMESEC	LLSMSITIMETONE	LSM/cmd/_/F.LLSMTIMESEC.shtml:%	Timestamp seconds
LLSMTIMESUB	LLSMSIANCILLARY	LSM/cmd/_/F.LLSMTIMESUB.shtml:%	Timestamp microseconds
LLSMTIMESUB	LLSMSIATTITUDE	LSM/cmd/_/F.LLSMTIMESUB.shtml:%	Timestamp microseconds
MSG_MsgLevel	msgResponse	LCM/cmd/_/F.MSG_MsgLevel.shtml:%	Messaging level
PAD16	LLFSDIRCREATE	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSDIRDELETE	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSDIRDUMP	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSFILECOPY	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSFILEDELETE	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSFILEDUMPC	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
PAD16	LLFSSYSSTATUS	LFS/cmd/_/F.PAD16.shtml:%	16-bit padding
Pad1	cmdResponse	LCM/cmd/_/F.Pad1.shtml:%	One byte padding
Pad1	msgResponse	LCM/cmd/_/F.Pad1.shtml:%	One byte padding
tem_mask	cal_lrs	LMC/cmd/_/F.tem_mask.shtml:%	TEM device mask
tem_mask	tkr_lrs	LMC/cmd/_/F.tem_mask.shtml:%	TEM device mask
tile_num	acd_tile_pair	LMC/cmd/_/F.tile_num.shtml:%	ACD Tile ID
tkr_mask	tkr_lrs	LMC/cmd/_/F.tkr_mask.shtml:%	TKR LRS Mask
WORD16	HtrOnOffCtl	LTC/cmd/_/F.WORD16.shtml:%	Unsigned 16-bit word
WORD16	SetMode	LTC/cmd/_/F.WORD16.shtml:%	Unsigned 16-bit word
WORD16	SetParam	LTC/cmd/_/F.WORD16.shtml:%	Unsigned 16-bit word
WORD16	SetTlmFreq	LTC/cmd/_/F.WORD16.shtml:%	Unsigned 16-bit word
WORD16	Start	LTC/cmd/_/F.WORD16.shtml:%	Unsigned 16-bit word

22 Telecommand Enumeration Index, by Name

Enumeration	Section	Description
CMD_CNT_SEL	ISIS/att/_/E.CMD_CNT_SEL.shtml:%	Selects count to return as diagnostic telemetry
EPU_ID	ISIS/att/_/E.EPU_ID.shtml:%	Enumeration of the EPUs
FILEDEVICE	ISIS/att/_/E.FILEDEVICE.shtml:%	Code for file device
ON_OFF_SELECTOR	ISIS/att/_/E.ON_OFF_SELECTOR.shtml:%	Enumeration of the options for the simple on-off selector
P_S_SELECTOR	ISIS/att/_/E.P_S_SELECTOR.shtml:%	Enumeration for the simple primary-secondary selector
PDU_ID	ISIS/att/_/E.PDU_ID.shtml:%	Identifies a PDU
SCIPATTYPE	ISIS/att/_/E.SCIPATTYPE.shtml:%	Science data generation pattern types
SIU_ID	ISIS/att/_/E.SIU_ID.shtml:%	Enumeration of the possible SIU IDs

23 Telecommand Range Index, by Name

Range	Section	Description
bits_12_range	ISIS/att/_/R.bits_12_range.shtml:%	Range for 12-bit fields
cmd_cnt_range	ISIS/att/_/R.cmd_cnt_range.shtml:%	Range for command count selection
epu_range	ISIS/att/_/R.epu_range.shtml:%	EPU number range
LHKAPIDRNG	LHK/att/_/R.LHKAPIDRNG.shtml:%	LHK APID Range
LHKDIAGINTV	LHK/att/_/R.LHKDIAGINTV.shtml:%	Diagnostic Interval
LHKDIAGPKTCNT	LHK/att/_/R.LHKDIAGPKTCNT.shtml:%	Diagnostic Packet Count
on_off_range	ISIS/att/_/R.on_off_range.shtml:%	On and off selector range
p_s_range	ISIS/att/_/R.p_s_range.shtml:%	Primary and secondary selector range
pdu_id_range	ISIS/att/_/R.pdu_id_range.shtml:%	PDU ID range
siu_id_range	ISIS/att/_/R.siu_id_range.shtml:%	SIU ID range
tem_mask_range	ISIS/att/_/R.tem_mask_range.shtml:%	TEM mask range

24 Telemetry Packet Index, by APID

APID	Tlm. Packet (L)	Section	Description (L)
0x200	LBTHKP	PBC/tlm/_/P.LBTHKP.shtml:%	Boot housekeeping telemetry
0x210	TemEnvPwr0	LHK/tlm/_/P.TemEnvPwr0.shtml:%	TEM Power Packet 0
0x211	TemEnvPwr1	LHK/tlm/_/P.TemEnvPwr1.shtml:%	TEM Power Packet 1
0x212	TemEnvPwr2	LHK/tlm/_/P.TemEnvPwr2.shtml:%	TEM Power Packet 2
0x213	TemEnvPwr3	LHK/tlm/_/P.TemEnvPwr3.shtml:%	TEM Power Packet 3
0x214	TemEnvPwr4	LHK/tlm/_/P.TemEnvPwr4.shtml:%	TEM Power Packet 4
0x215	TemEnvPwr5	LHK/tlm/_/P.TemEnvPwr5.shtml:%	TEM Power Packet 5
0x216	TemEnvTemp0	LHK/tlm/_/P.TemEnvTemp0.shtml:%	TEM Temperature Packet 0
0x217	TemEnvTemp1	LHK/tlm/_/P.TemEnvTemp1.shtml:%	TEM Temperature Packet 1
0x218	TemEnvTemp2	LHK/tlm/_/P.TemEnvTemp2.shtml:%	TEM Temperature Packet 2
0x219	TemEnvTemp3	LHK/tlm/_/P.TemEnvTemp3.shtml:%	TEM Temperature Packet 3
0x21A	TemEnvTemp4	LHK/tlm/_/P.TemEnvTemp4.shtml:%	TEM Temperature Packet 4
[illegible]	[illegible]	[illegible]	TEM Temperature Packet 5
[illegible]	[illegible]	[illegible]	TEM Temperature Packet 6
[illegible]	[illegible]	[illegible]	TEM Temperature Packet 7
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 0
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 1
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 2
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 3
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 4
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 5
[illegible]	[illegible]	[illegible]	PDU Environmental Packet 6
[illegible]	[illegible]	[illegible]	PDU Environmental Packet [illegible]
[illegible]	[illegible]	[illegible]	AEM Environmental Monitor Packet 0
[illegible]	[illegible]	[illegible]	Low-rate Science Packet
[illegible]	[illegible]	[illegible]	Command Statistics Packet 0
[illegible]	[illegible]	[illegible]	Command Statistics Packet 1
[illegible]	[illegible]	[illegible]	File System Statistics Packet
[illegible]	[illegible]	[illegible]	CPU Metrics/RT Statistics Packet
[illegible]	[illegible]	[illegible]	Memory Load/Dump Statistics
[illegible]	[illegible]	[illegible]	Memory Load/Dump Statistics
[illegible]	[illegible]	[illegible]	EPU 0 Boot Housekeeping Telemetry
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 0
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 1
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 2
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 3
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 4
[illegible]	[illegible]	[illegible]	Diagnostic TEM Power Packet 5
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet 0
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet 1
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic TEM Temperature Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic PDU Environmental Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic PDU Environmental Packet [illegible]
[illegible]	[illegible]	[illegible]	Diagnostic PDU Environmental Packet 2
[illegible]	[illegible]	[illegible]	Diagnostic PDU Environmental Packet 3
[illegible]	[illegible]	[illegible]	Diagnostic PDU Environmental Packet 4

APID	Tlm Packet (L)	Section	Description (R)
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Environmental Packet 5
[illegible]	[illegible]	[illegible]	Diagnostic P[illegible] Environmental Packet 6
[illegible]	[illegible]	[illegible]	Diagnostic P[illegible] Environmental Packet 7
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Environmental Monitor Packet 0
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Science Packet
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Statistics Packet
[illegible]	[illegible]	[illegible]	[illegible] Statistics [illegible]
[illegible]	[illegible]	[illegible]	[illegible] Packet
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Statistics Packet
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Dump Statistics
[illegible]	[illegible]	[illegible]	Diagnostic [illegible] Dump Statistics
[illegible]	[illegible]	[illegible]	[illegible] Thermal Control Diagnostic Telemetry
[illegible]	[illegible]	[illegible]	[illegible] Low [illegible] Science Counters
[illegible]	[illegible]	[illegible]	[illegible] Low [illegible] Science Counters
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible] Value [illegible]
[illegible]	[illegible]	[illegible]	[illegible] Data
[illegible]	[illegible]	[illegible]	[illegible] Report
[illegible]	[illegible]	[illegible]	Re[illegible] Report
[illegible]	[illegible]	[illegible]	File [illegible] Status Report
[illegible]	[illegible]	[illegible]	File [illegible] CTDB
[illegible]	[illegible]	[illegible]	[illegible] Limit Alert Packet

25 Telemetry Packet Index, by Name (LCAT)

Tlm. Packet (L)	APID	Section	Description (L)
acd_cnt	0x2C3	LMC/tlm/_/P.acd_cnt.shtml:%	ACD Tile Counters
AemEnv0	0x226	LHK/tlm/_/P.AemEnv0.shtml:%	AEM Environmental Monitor Packet 0
cal_cnt	0x2C1	LMC/tlm/_/P.cal_cnt.shtml:%	CAL Low Rate Science Counters
CmdCnt0	0x228	LHK/tlm/_/P.CmdCnt0.shtml:%	Command Statistics Packet 0
CmdCnt1	0x229	LHK/tlm/_/P.CmdCnt1.shtml:%	Command Statistics Packet 1
CmdConfirm	0x2D0	ITC/tlm/_/P.CmdConfirm.shtml:%	Response to command from spacecraft
CpuMetr	0x22B	LHK/tlm/_/P.CpuMetr.shtml:%	CPU Metrics/RT Statistics Packet
DiagAemEnv0	0x286	LHK/tlm/_/P.DiagAemEnv0.shtml:%	Diagnostic AEM Environmental Monitor Packet 0
DiagCmdCnt0	0x288	LHK/tlm/_/P.DiagCmdCnt0.shtml:%	Diagnostic Command Statistics Packet 0
DiagCmdCnt1	0x289	LHK/tlm/_/P.DiagCmdCnt1.shtml:%	Diagnostic Command Statistics Packet 1
DiagCpuMetr	0x28B	LHK/tlm/_/P.DiagCpuMetr.shtml:%	Diagnostic CPU Metrics/RT Statistics Packet
DiagFileStats	0x28A	LHK/tlm/_/P.DiagFileStats.shtml:%	Diagnositc File System Statistics Packet
DiagLrs0	0x287	LHK/tlm/_/P.DiagLrs0.shtml:%	Diagnostic Low-rate Science Packet
DiagLTC	0x2BC	LTC/tlm/_/P.DiagLTC.shtml:%	LAT Thermal Control diagnostic telemetry
DiagMemStats0	0x28C	LHK/tlm/_/P.DiagMemStats0.shtml:%	Diagnostic Memory Load/Dump Statistics
DiagMemStats1	0x28D	LHK/tlm/_/P.DiagMemStats1.shtml:%	Diagnostic Memory Load/Dump Statistics
DiagPduEnv0	0x27E	LHK/tlm/_/P.DiagPduEnv0.shtml:%	Diagnostic PDU Environmental Packet 0
DiagPduEnv1	0x27F	LHK/tlm/_/P.DiagPduEnv1.shtml:%	Diagnostic PDU Environmental Packet 1
DiagPduEnv2	0x280	LHK/tlm/_/P.DiagPduEnv2.shtml:%	Diagnostic PDU Environmental Packet 2
DiagPduEnv3	0x281	LHK/tlm/_/P.DiagPduEnv3.shtml:%	Diagnostic PDU Environmental Packet 3
DiagPduEnv4	0x282	LHK/tlm/_/P.DiagPduEnv4.shtml:%	Diagnostic PDU Environmental Packet 4
DiagPduEnv5	0x283	LHK/tlm/_/P.DiagPduEnv5.shtml:%	Diagnostic PDU Environmental Packet 5
DiagPduEnv6	0x284	LHK/tlm/_/P.DiagPduEnv6.shtml:%	Diagnostic PDU Environmental Packet 6
DiagPduEnv7	0x285	LHK/tlm/_/P.DiagPduEnv7.shtml:%	Diagnostic PDU Environmental Packet 7
DiagTemEnvPwr0	0x270	LHK/tlm/_/P.DiagTemEnvPwr0.shtml:%	Diagnostic TEM Power Packet 0
DiagTemEnvPwr1	0x271	LHK/tlm/_/P.DiagTemEnvPwr1.shtml:%	Diagnostic TEM Power Packet 1
DiagTemEnvPwr2	0x272	LHK/tlm/_/P.DiagTemEnvPwr2.shtml:%	Diagnostic TEM Power Packet 2
DiagTemEnvPwr3	0x273	LHK/tlm/_/P.DiagTemEnvPwr3.shtml:%	Diagnostic TEM Power Packet 3
DiagTemEnvPwr4	0x274	LHK/tlm/_/P.DiagTemEnvPwr4.shtml:%	Diagnostic TEM Power Packet 4
DiagTemEnvPwr5	0x275	LHK/tlm/_/P.DiagTemEnvPwr5.shtml:%	Diagnostic TEM Power Packet 5
DiagTemEnvTemp0	0x276	LHK/tlm/_/P.DiagTemEnvTemp0.shtml:%	Diagnostic TEM Temperature Packet 0
DiagTemEnvTemp1	0x277	LHK/tlm/_/P.DiagTemEnvTemp1.shtml:%	Diagnostic TEM Temperature Packet 1
DiagTemEnvTemp2	0x278	LHK/tlm/_/P.DiagTemEnvTemp2.shtml:%	Diagnostic TEM Temperature Packet 2
DiagTemEnvTemp3	0x279	LHK/tlm/_/P.DiagTemEnvTemp3.shtml:%	Diagnostic TEM Temperature Packet 3
DiagTemEnvTemp4	0x27A	LHK/tlm/_/P.DiagTemEnvTemp4.shtml:%	Diagnostic TEM Temperature Packet 4
DiagTemEnvTemp5	0x27B	LHK/tlm/_/P.DiagTemEnvTemp5.shtml:%	Diagnostic TEM Temperature Packet 5
DiagTemEnvTemp6	0x27C	LHK/tlm/_/P.DiagTemEnvTemp6.shtml:%	Diagnostic TEM Temperature Packet 6
DiagTemEnvTemp7	0x27D	LHK/tlm/_/P.DiagTemEnvTemp7.shtml:%	Diagnostic TEM Temperature Packet 7
FileStats	0x22A	LHK/tlm/_/P.FileStats.shtml:%	File System Statistics Packet
LBTEPU0HKP	0x261	PBC/tlm/_/P.LBTEPU0HKP.shtml:%	EPU 0 Boot Housekeeping Telemetry
LBTHKP	0x200	PBC/tlm/_/P.LBTHKP.shtml:%	Boot housekeeping telemetry
LLFSDIRLIST	0x318	LFS/tlm/_/P.LLFSDIRLIST.shtml:%	Directory Listing Report
LLFSDUMPCTDB	0x31B	LFS/tlm/_/P.LLFSDUMPCTDB.shtml:%	File Dump Data CTDB
LLFSROOTLIST	0x319	LFS/tlm/_/P.LLFSROOTLIST.shtml:%	Root Listing Report
LLFSSYSLIST	0x31A	LFS/tlm/_/P.LLFSSYSLIST.shtml:%	File System Status Report
LMEMPOOLDATA	0x311	MEM/tlm/_/P.LMEMPOOLDATA.shtml:%	Memory Pool Statistics Dump
LMEMSIUDATA	0x314	MEM/tlm/_/P.LMEMSIUDATA.shtml:%	SIU Memory Dump Data
LMEMSYMVAL	0x312	MEM/tlm/_/P.LMEMSYMVAL.shtml:%	Symbol Value Dump
Lrs0	0x227	LHK/tlm/_/P.Lrs0.shtml:%	Low-rate Science Packet
MemStats0	0x22C	LHK/tlm/_/P.MemStats0.shtml:%	Memory Load/Dump Statistics
MemStats1	0x22D	LHK/tlm/_/P.MemStats1.shtml:%	Memory Load/Dump Statistics

Tlm. Packet (L)	APID	Section	Description (L)
PduEnv0	0x21E	LHK/tlm/_/P.PduEnv0.shtml:%	PDU Environmental Packet 0
PduEnv1	0x21F	LHK/tlm/_/P.PduEnv1.shtml:%	PDU Environmental Packet 1
PduEnv2	0x220	LHK/tlm/_/P.PduEnv2.shtml:%	PDU Environmental Packet 2
PduEnv3	0x221	LHK/tlm/_/P.PduEnv3.shtml:%	PDU Environmental Packet 3
PduEnv4	0x222	LHK/tlm/_/P.PduEnv4.shtml:%	PDU Environmental Packet 4
PduEnv5	0x223	LHK/tlm/_/P.PduEnv5.shtml:%	PDU Environmental Packet 5
PduEnv6	0x224	LHK/tlm/_/P.PduEnv6.shtml:%	PDU Environmental Packet 6
PduEnv7	0x225	LHK/tlm/_/P.PduEnv7.shtml:%	PDU Environmental Packet 7
RedLimAlrt	0x353	LHK/tlm/_/P.RedLimAlrt.shtml:%	Red Limit Alert Packet
TemEnvPwr0	0x210	LHK/tlm/_/P.TemEnvPwr0.shtml:%	TEM Power Packet 0
TemEnvPwr1	0x211	LHK/tlm/_/P.TemEnvPwr1.shtml:%	TEM Power Packet 1
TemEnvPwr2	0x212	LHK/tlm/_/P.TemEnvPwr2.shtml:%	TEM Power Packet 2
TemEnvPwr3	0x213	LHK/tlm/_/P.TemEnvPwr3.shtml:%	TEM Power Packet 3
TemEnvPwr4	0x214	LHK/tlm/_/P.TemEnvPwr4.shtml:%	TEM Power Packet 4
TemEnvPwr5	0x215	LHK/tlm/_/P.TemEnvPwr5.shtml:%	TEM Power Packet 5
TemEnvTemp0	0x216	LHK/tlm/_/P.TemEnvTemp0.shtml:%	TEM Temperature Packet 0
TemEnvTemp1	0x217	LHK/tlm/_/P.TemEnvTemp1.shtml:%	TEM Temperature Packet 1
TemEnvTemp2	0x218	LHK/tlm/_/P.TemEnvTemp2.shtml:%	TEM Temperature Packet 2
TemEnvTemp3	0x219	LHK/tlm/_/P.TemEnvTemp3.shtml:%	TEM Temperature Packet 3
TemEnvTemp4	0x21A	LHK/tlm/_/P.TemEnvTemp4.shtml:%	TEM Temperature Packet 4
TemEnvTemp5	0x21B	LHK/tlm/_/P.TemEnvTemp5.shtml:%	TEM Temperature Packet 5
TemEnvTemp6	0x21C	LHK/tlm/_/P.TemEnvTemp6.shtml:%	TEM Temperature Packet 6
TemEnvTemp7	0x21D	LHK/tlm/_/P.TemEnvTemp7.shtml:%	TEM Temperature Packet 7
tkr_cnt	0x2C2	LMC/tlm/_/P.tkr_cnt.shtml:%	TKR Low Rate Science Counters

26 Telemetry Struct Index, by Name (LCAT)

Struct (L)	Tlm. Packet (L)	Section	Description (L)
AEMDAQENV	DiagPduEnv0	LHK/tlm/_/S.AEMDAQENV.shtml:%	AEM DAQ Environmental
AEMDAQENV	PduEnv0	LHK/tlm/_/S.AEMDAQENV.shtml:%	AEM DAQ Environmental
AEMFRENV	AemEnv0	LHK/tlm/_/S.AEMFRENV.shtml:%	AEM Free Board Environmental ADCs
AEMFRENV	DiagAemEnv0	LHK/tlm/_/S.AEMFRENV.shtml:%	AEM Free Board Environmental ADCs
cal_counter	cal_cnt	LMC/tlm/_/S.cal_counter.shtml:%	CAL LRS Counter Structure
CMDCNTRS	CmdCnt0	LHK/tlm/_/S.CMDCNTRS.shtml:%	Command Counters
CMDCNTRS	CmdCnt1	LHK/tlm/_/S.CMDCNTRS.shtml:%	Command Counters
CMDCNTRS	DiagCmdCnt0	LHK/tlm/_/S.CMDCNTRS.shtml:%	Command Counters
CMDCNTRS	DiagCmdCnt1	LHK/tlm/_/S.CMDCNTRS.shtml:%	Command Counters
CmdHeader	CmdConfirm	ITC/tlm/_/S.CmdHeader.shtml:%	CCSDS telecommand header
CPUMETR	CpuMetr	LHK/tlm/_/S.CPUMETR.shtml:%	CPU Metrics
CPUMETR	DiagCpuMetr	LHK/tlm/_/S.CPUMETR.shtml:%	CPU Metrics
FILESTATS	DiagFileStats	LHK/tlm/_/S.FILESTATS.shtml:%	FILE Upload Statistics
FILESTATS	FileStats	LHK/tlm/_/S.FILESTATS.shtml:%	FILE Upload Statistics
GEMLRS	DiagLrs0	LHK/tlm/_/S.GEMLRS.shtml:%	GEM Low-Rate Science Counter
GEMLRS	Lrs0	LHK/tlm/_/S.GEMLRS.shtml:%	GEM Low-Rate Science Counter
MEMSTATS	DiagMemStats0	LHK/tlm/_/S.MEMSTATS.shtml:%	MEM Statistics
MEMSTATS	DiagMemStats1	LHK/tlm/_/S.MEMSTATS.shtml:%	MEM Statistics
MEMSTATS	MemStats0	LHK/tlm/_/S.MEMSTATS.shtml:%	MEM Statistics
MEMSTATS	MemStats1	LHK/tlm/_/S.MEMSTATS.shtml:%	MEM Statistics
RTSTATS	CpuMetr	LHK/tlm/_/S.RTSTATS.shtml:%	1553 Remote Terminal Statistics
RTSTATS	DiagCpuMetr	LHK/tlm/_/S.RTSTATS.shtml:%	1553 Remote Terminal Statistics
tile_counter	acd_cnt	LMC/tlm/_/S.tile_counter.shtml:%	ADC Tile Counter Structure
tkr_counter	tkr_cnt	LMC/tlm/_/S.tkr_counter.shtml:%	TKR LRS Counter Structure
TMUX0	DiagTemEnvPwr0	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	DiagTemEnvPwr1	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	DiagTemEnvPwr2	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	DiagTemEnvPwr3	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	DiagTemEnvPwr4	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	DiagTemEnvPwr5	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr0	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr1	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr2	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr3	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr4	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX0	TemEnvPwr5	LHK/tlm/_/S.TMUX0.shtml:%	TEM mux channel 0 environmental ADCs
TMUX1	DiagTemEnvPwr0	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	DiagTemEnvPwr1	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	DiagTemEnvPwr2	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	DiagTemEnvPwr3	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	DiagTemEnvPwr4	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	DiagTemEnvPwr5	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr0	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr1	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr2	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr3	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr4	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX1	TemEnvPwr5	LHK/tlm/_/S.TMUX1.shtml:%	TEM mux channel 1 environmental ADCs
TMUX2	DiagTemEnvTemp0	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp1	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp2	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs

Struct (L)	Tlm. Packet (L)	Section	Description (L)
TMUX2	DiagTemEnvTemp3	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp4	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp5	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp6	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	DiagTemEnvTemp7	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp0	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp1	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp2	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp3	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp4	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp5	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp6	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX2	TemEnvTemp7	LHK/tlm/_/S.TMUX2.shtml:%	TEM mux channel 2 environmental ADCs
TMUX3	DiagTemEnvTemp0	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp1	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp2	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp3	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp4	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp5	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp6	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	DiagTemEnvTemp7	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp0	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp1	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp2	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp3	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp4	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp5	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp6	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX3	TemEnvTemp7	LHK/tlm/_/S.TMUX3.shtml:%	TEM mux channel 3 environmental ADCs
TMUX4	DiagTemEnvTemp0	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp1	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp2	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp3	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp4	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp5	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp6	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	DiagTemEnvTemp7	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp0	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp1	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp2	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp3	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp4	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp5	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp6	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs
TMUX4	TemEnvTemp7	LHK/tlm/_/S.TMUX4.shtml:%	TEM mux channel 4 environmental ADCs

27 Telemetry Bitfield Index, by Name (LCAT)

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
ACDBEAGRIDTEMP	DiagPduEnv3	LHK/tlm/_/B.ACDBEAGRIDTEMP.shtml:%	ACD BEA Grid Interface Tempe
ACDBEAGRIDTEMP	DiagPduEnv7	LHK/tlm/_/B.ACDBEAGRIDTEMP.shtml:%	ACD BEA Grid Interface Tempe
ACDBEAGRIDTEMP	PduEnv3	LHK/tlm/_/B.ACDBEAGRIDTEMP.shtml:%	ACD BEA Grid Interface Tempe
ACDBEAGRIDTEMP	PduEnv7	LHK/tlm/_/B.ACDBEAGRIDTEMP.shtml:%	ACD BEA Grid Interface Tempe
ACDPMTRAILTEMP	DiagPduEnv3	LHK/tlm/_/B.ACDPMTRAILTEMP.shtml:%	ADC PMT Rail Temperature
ACDPMTRAILTEMP	DiagPduEnv7	LHK/tlm/_/B.ACDPMTRAILTEMP.shtml:%	ADC PMT Rail Temperature
ACDPMTRAILTEMP	PduEnv3	LHK/tlm/_/B.ACDPMTRAILTEMP.shtml:%	ADC PMT Rail Temperature
ACDPMTRAILTEMP	PduEnv7	LHK/tlm/_/B.ACDPMTRAILTEMP.shtml:%	ADC PMT Rail Temperature
ACDPWRREG	DiagPduEnv0	LHK/tlm/_/B.ACDPWRREG.shtml:%	PDU ADC Power Management R
ACDPWRREG	DiagPduEnv4	LHK/tlm/_/B.ACDPWRREG.shtml:%	PDU ADC Power Management R
ACDPWRREG	PduEnv0	LHK/tlm/_/B.ACDPWRREG.shtml:%	PDU ADC Power Management R
ACDPWRREG	PduEnv4	LHK/tlm/_/B.ACDPWRREG.shtml:%	PDU ADC Power Management R
ACDSHELLTEMP	DiagPduEnv3	LHK/tlm/_/B.ACDSHELLTEMP.shtml:%	ADC Shell Temperature
ACDSHELLTEMP	DiagPduEnv7	LHK/tlm/_/B.ACDSHELLTEMP.shtml:%	ADC Shell Temperature
ACDSHELLTEMP	PduEnv3	LHK/tlm/_/B.ACDSHELLTEMP.shtml:%	ADC Shell Temperature
ACDSHELLTEMP	PduEnv7	LHK/tlm/_/B.ACDSHELLTEMP.shtml:%	ADC Shell Temperature
AEMFR28ISUM	DiagPduEnv0	LHK/tlm/_/B.AEMFR28ISUM.shtml:%	AEM FREE Board 28V Sum
AEMFR28ISUM	PduEnv0	LHK/tlm/_/B.AEMFR28ISUM.shtml:%	AEM FREE Board 28V Sum
AEMFR33ISUM	DiagPduEnv0	LHK/tlm/_/B.AEMFR33ISUM.shtml:%	AEM FREE Board 3.3 Sum
AEMFR33ISUM	PduEnv0	LHK/tlm/_/B.AEMFR33ISUM.shtml:%	AEM FREE Board 3.3 Sum
AEMFRHV1	AemEnv0	LHK/tlm/_/B.AEMFRHV1.shtml:%	AEM Free Board HV1
AEMFRHV1	DiagAemEnv0	LHK/tlm/_/B.AEMFRHV1.shtml:%	AEM Free Board HV1
AEMFRHV2	AemEnv0	LHK/tlm/_/B.AEMFRHV2.shtml:%	AEM Free Board HV2
AEMFRHV2	DiagAemEnv0	LHK/tlm/_/B.AEMFRHV2.shtml:%	AEM Free Board HV2
AEMFRPWRREG	AemEnv0	LHK/tlm/_/B.AEMFRPWRREG.shtml:%	AEM FREE board power status r
AEMFRPWRREG	DiagAemEnv0	LHK/tlm/_/B.AEMFRPWRREG.shtml:%	AEM FREE board power status r
AEMFRTEMP	AemEnv0	LHK/tlm/_/B.AEMFRTEMP.shtml:%	AEM Free board Temperature
AEMFRTEMP	DiagAemEnv0	LHK/tlm/_/B.AEMFRTEMP.shtml:%	AEM Free board Temperature
AEMFRVDD	AemEnv0	LHK/tlm/_/B.AEMFRVDD.shtml:%	AEM VDD
AEMFRVDD	DiagAemEnv0	LHK/tlm/_/B.AEMFRVDD.shtml:%	AEM VDD
AFEETEMP	DiagTemEnvTemp0	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp1	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp2	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp3	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp4	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp5	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp6	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	DiagTemEnvTemp7	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp0	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp1	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp2	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp3	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp4	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp5	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp6	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
AFEETEMP	TemEnvTemp7	LHK/tlm/_/B.AFEETEMP.shtml:%	CAL AFFE temperature ADC va
BOOTCOMMAND	LBTEPU0HKP	PBC/tlm/_/B.BOOTCOMMAND.shtml:%	Last boot command bitfield
BOOTCOMMAND	LBTHKP	PBC/tlm/_/B.BOOTCOMMAND.shtml:%	Last boot command bitfield
CAL33I	DiagTemEnvPwr0	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	DiagTemEnvPwr1	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	DiagTemEnvPwr2	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
CAL33I	DiagTemEnvPwr3	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	DiagTemEnvPwr4	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	DiagTemEnvPwr5	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr0	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr1	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr2	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr3	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr4	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33I	TemEnvPwr5	LHK/tlm/_/B.CAL33I.shtml:%	CAL 3.3 Current ADC value
CAL33V	DiagTemEnvPwr0	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	DiagTemEnvPwr1	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	DiagTemEnvPwr2	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	DiagTemEnvPwr3	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	DiagTemEnvPwr4	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	DiagTemEnvPwr5	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr0	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr1	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr2	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr3	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr4	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CAL33V	TemEnvPwr5	LHK/tlm/_/B.CAL33V.shtml:%	CAL 3.3 Voltage ADC value
CALBIASI	DiagTemEnvPwr0	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	DiagTemEnvPwr1	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	DiagTemEnvPwr2	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	DiagTemEnvPwr3	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	DiagTemEnvPwr4	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	DiagTemEnvPwr5	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr0	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr1	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr2	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr3	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr4	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASI	TemEnvPwr5	LHK/tlm/_/B.CALBIASI.shtml:%	CAL Bias Current ADC value
CALBIASV	DiagTemEnvPwr0	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	DiagTemEnvPwr1	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	DiagTemEnvPwr2	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	DiagTemEnvPwr3	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	DiagTemEnvPwr4	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	DiagTemEnvPwr5	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr0	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr1	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr2	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr3	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr4	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBIASV	TemEnvPwr5	LHK/tlm/_/B.CALBIASV.shtml:%	CAL Bias Voltage ADC value
CALBSPTEMP	DiagPduEnv1	LHK/tlm/_/B.CALBSPTEMP.shtml:%	CAL Baseplate Temperature
CALBSPTEMP	DiagPduEnv5	LHK/tlm/_/B.CALBSPTEMP.shtml:%	CAL Baseplate Temperature
CALBSPTEMP	PduEnv1	LHK/tlm/_/B.CALBSPTEMP.shtml:%	CAL Baseplate Temperature
CALBSPTEMP	PduEnv5	LHK/tlm/_/B.CALBSPTEMP.shtml:%	CAL Baseplate Temperature
DABTEMP	DiagPduEnv0	LHK/tlm/_/B.DABTEMP.shtml:%	DAQ Board Temperature
DABTEMP	PduEnv0	LHK/tlm/_/B.DABTEMP.shtml:%	DAQ Board Temperature
DAQ33V	DiagPduEnv0	LHK/tlm/_/B.DAQ33V.shtml:%	DAQ Board 3.3 Voltage

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
DAQ33V	PduEnv0	LHK/tlm/_/B.DAQ33V.shtml:%	DAQ Board 3.3 Voltage
DUMPSTFLAGS	LLFSDIRLIST	LFS/tlm/_/B.DUMPSTFLAGS.shtml:%	File Dump Flags
DUMPSTFLAGS	LLFSDUMPCTDB	LFS/tlm/_/B.DUMPSTFLAGS.shtml:%	File Dump Flags
DUMPSTFLAGS	LLFSROOTLIST	LFS/tlm/_/B.DUMPSTFLAGS.shtml:%	File Dump Flags
DUMPSTFLAGS	LLFSSYSLIST	LFS/tlm/_/B.DUMPSTFLAGS.shtml:%	File Dump Flags
EPUTEMP	DiagPduEnv0	LHK/tlm/_/B.EPUTEMP.shtml:%	EPU Temperature ADC value
EPUTEMP	DiagPduEnv4	LHK/tlm/_/B.EPUTEMP.shtml:%	EPU Temperature ADC value
EPUTEMP	PduEnv0	LHK/tlm/_/B.EPUTEMP.shtml:%	EPU Temperature ADC value
EPUTEMP	PduEnv4	LHK/tlm/_/B.EPUTEMP.shtml:%	EPU Temperature ADC value
EPUV	DiagPduEnv0	LHK/tlm/_/B.EPUV.shtml:%	EPU Voltage ADC value
EPUV	DiagPduEnv4	LHK/tlm/_/B.EPUV.shtml:%	EPU Voltage ADC value
EPUV	PduEnv0	LHK/tlm/_/B.EPUV.shtml:%	EPU Voltage ADC value
EPUV	PduEnv4	LHK/tlm/_/B.EPUV.shtml:%	EPU Voltage ADC value
FILESTFLAGS	LLFSDIRLIST	LFS/tlm/_/B.FILESTFLAGS.shtml:%	File Storage Flags
FILESTFLAGS	LLFSROOTLIST	LFS/tlm/_/B.FILESTFLAGS.shtml:%	File Storage Flags
FILESTID	LLFSDIRLIST	LFS/tlm/_/B.FILESTID.shtml:%	File Storage ID
FILESTID	LLFSDUMPCTDB	LFS/tlm/_/B.FILESTID.shtml:%	File Storage ID
FILESTID	LLFSROOTLIST	LFS/tlm/_/B.FILESTID.shtml:%	File Storage ID
FILESTID	LLFSSYSLIST	LFS/tlm/_/B.FILESTID.shtml:%	File Storage ID
GRIDRADIFTEMP	DiagPduEnv3	LHK/tlm/_/B.GRIDRADIFTEMP.shtml:%	Grid Radiator Interface Temperat
GRIDRADIFTEMP	DiagPduEnv7	LHK/tlm/_/B.GRIDRADIFTEMP.shtml:%	Grid Radiator Interface Temperat
GRIDRADIFTEMP	PduEnv3	LHK/tlm/_/B.GRIDRADIFTEMP.shtml:%	Grid Radiator Interface Temperat
GRIDRADIFTEMP	PduEnv7	LHK/tlm/_/B.GRIDRADIFTEMP.shtml:%	Grid Radiator Interface Temperat
GRIDTEMP	DiagPduEnv3	LHK/tlm/_/B.GRIDTEMP.shtml:%	Grid Temperature
GRIDTEMP	DiagPduEnv7	LHK/tlm/_/B.GRIDTEMP.shtml:%	Grid Temperature
GRIDTEMP	PduEnv3	LHK/tlm/_/B.GRIDTEMP.shtml:%	Grid Temperature
GRIDTEMP	PduEnv7	LHK/tlm/_/B.GRIDTEMP.shtml:%	Grid Temperature
Hdr0	CmdConfirm	ITC/tlm/_/B.Hdr0.shtml:%	CCSDS telecommand header (by
Hdr1	CmdConfirm	ITC/tlm/_/B.Hdr1.shtml:%	CCSDS telecommand header (by
Hdr3	CmdConfirm	ITC/tlm/_/B.Hdr3.shtml:%	CCSDS telecommand header (by
PDUEPUCRATEPWR	DiagPduEnv0	LHK/tlm/_/B.PDUEPUCRATEPWR.shtml:%	PDU EPU Crate Power States
PDUEPUCRATEPWR	DiagPduEnv4	LHK/tlm/_/B.PDUEPUCRATEPWR.shtml:%	PDU EPU Crate Power States
PDUEPUCRATEPWR	PduEnv0	LHK/tlm/_/B.PDUEPUCRATEPWR.shtml:%	PDU EPU Crate Power States
PDUEPUCRATEPWR	PduEnv4	LHK/tlm/_/B.PDUEPUCRATEPWR.shtml:%	PDU EPU Crate Power States
PDUTEMPWRREG	DiagPduEnv0	LHK/tlm/_/B.PDUTEMPWRREG.shtml:%	PDU TEM Power Management R
PDUTEMPWRREG	DiagPduEnv4	LHK/tlm/_/B.PDUTEMPWRREG.shtml:%	PDU TEM Power Management R
PDUTEMPWRREG	PduEnv0	LHK/tlm/_/B.PDUTEMPWRREG.shtml:%	PDU TEM Power Management R
PDUTEMPWRREG	PduEnv4	LHK/tlm/_/B.PDUTEMPWRREG.shtml:%	PDU TEM Power Management R
RADANHTRTEMP	DiagPduEnv3	LHK/tlm/_/B.RADANHTRTEMP.shtml:%	Radiator Anitfreeze Heater Temp
RADANHTRTEMP	DiagPduEnv7	LHK/tlm/_/B.RADANHTRTEMP.shtml:%	Radiator Anitfreeze Heater Temp
RADANHTRTEMP	PduEnv3	LHK/tlm/_/B.RADANHTRTEMP.shtml:%	Radiator Anitfreeze Heater Temp
RADANHTRTEMP	PduEnv7	LHK/tlm/_/B.RADANHTRTEMP.shtml:%	Radiator Anitfreeze Heater Temp
RADTEMP	DiagPduEnv3	LHK/tlm/_/B.RADTEMP.shtml:%	Radiator Temperature
RADTEMP	DiagPduEnv7	LHK/tlm/_/B.RADTEMP.shtml:%	Radiator Temperature
RADTEMP	PduEnv3	LHK/tlm/_/B.RADTEMP.shtml:%	Radiator Temperature
RADTEMP	PduEnv7	LHK/tlm/_/B.RADTEMP.shtml:%	Radiator Temperature
ResRitSnsrSel	DiagLTC	LTC/tlm/_/B.ResRitSnsrSel.shtml:%	Selected RSVR and RIT sensors
ResRitStatus	DiagLTC	LTC/tlm/_/B.ResRitStatus.shtml:%	Heat pipe RIT and reservoir senso
TEM33I	DiagTemEnvPwr0	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	DiagTemEnvPwr1	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	DiagTemEnvPwr2	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	DiagTemEnvPwr3	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
TEM33I	DiagTemEnvPwr4	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	DiagTemEnvPwr5	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr0	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr1	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr2	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr3	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr4	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33I	TemEnvPwr5	LHK/tlm/_/B.TEM33I.shtml:%	TEM 3.3 Current ADC value
TEM33V	DiagPduEnv0	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagPduEnv4	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr0	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr1	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr2	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr3	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr4	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	DiagTemEnvPwr5	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	PduEnv0	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	PduEnv4	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr0	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr1	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr2	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr3	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr4	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEM33V	TemEnvPwr5	LHK/tlm/_/B.TEM33V.shtml:%	TEM 3.3 Voltage ADC value
TEMDEADTIMEREG	DiagLrs0	LHK/tlm/_/B.TEMDEADTIMEREG.shtml:%	TEM Deadtime Register Bitfield
TEMDEADTIMEREG	Lrs0	LHK/tlm/_/B.TEMDEADTIMEREG.shtml:%	TEM Deadtime Register Bitfield
TEMPCBTEMP	DiagPduEnv1	LHK/tlm/_/B.TEMPCBTEMP.shtml:%	TEM PCB Temperature
TEMPCBTEMP	DiagPduEnv5	LHK/tlm/_/B.TEMPCBTEMP.shtml:%	TEM PCB Temperature
TEMPCBTEMP	PduEnv1	LHK/tlm/_/B.TEMPCBTEMP.shtml:%	TEM PCB Temperature
TEMPCBTEMP	PduEnv5	LHK/tlm/_/B.TEMPCBTEMP.shtml:%	TEM PCB Temperature
TEMPSTEMP	DiagPduEnv0	LHK/tlm/_/B.TEMPSTEMP.shtml:%	TEM Power Supply Temperature
TEMPSTEMP	DiagPduEnv4	LHK/tlm/_/B.TEMPSTEMP.shtml:%	TEM Power Supply Temperature
TEMPSTEMP	PduEnv0	LHK/tlm/_/B.TEMPSTEMP.shtml:%	TEM Power Supply Temperature
TEMPSTEMP	PduEnv4	LHK/tlm/_/B.TEMPSTEMP.shtml:%	TEM Power Supply Temperature
TIDBF	LMEMPOOLDATA	MEM/tlm/_/B.TIDBF.shtml:%	LAT Unit and Transaction ID Bit
TIDBF	LMEMSIUDATA	MEM/tlm/_/B.TIDBF.shtml:%	LAT Unit and Transaction ID Bit
TIDBF	LMEMSYMVAL	MEM/tlm/_/B.TIDBF.shtml:%	LAT Unit and Transaction ID Bit
TKR15I	DiagTemEnvPwr0	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	DiagTemEnvPwr1	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	DiagTemEnvPwr2	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	DiagTemEnvPwr3	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	DiagTemEnvPwr4	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	DiagTemEnvPwr5	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr0	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr1	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr2	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr3	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr4	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15I	TemEnvPwr5	LHK/tlm/_/B.TKR15I.shtml:%	TKR 1.5 Current ADC value
TKR15V	DiagTemEnvPwr0	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	DiagTemEnvPwr1	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	DiagTemEnvPwr2	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
TKR15V	DiagTemEnvPwr3	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	DiagTemEnvPwr4	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	DiagTemEnvPwr5	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr0	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr1	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr2	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr3	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr4	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR15V	TemEnvPwr5	LHK/tlm/_/B.TKR15V.shtml:%	TKR 1.5 Voltage ADC value
TKR25I	DiagTemEnvPwr0	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	DiagTemEnvPwr1	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	DiagTemEnvPwr2	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	DiagTemEnvPwr3	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	DiagTemEnvPwr4	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	DiagTemEnvPwr5	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr0	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr1	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr2	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr3	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr4	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25I	TemEnvPwr5	LHK/tlm/_/B.TKR25I.shtml:%	TKR 2.5 Current ADC value
TKR25V	DiagTemEnvPwr0	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	DiagTemEnvPwr1	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	DiagTemEnvPwr2	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	DiagTemEnvPwr3	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	DiagTemEnvPwr4	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	DiagTemEnvPwr5	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr0	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr1	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr2	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr3	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr4	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKR25V	TemEnvPwr5	LHK/tlm/_/B.TKR25V.shtml:%	TKR 2.5 Voltage ADC value
TKRBIASI	DiagTemEnvPwr0	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	DiagTemEnvPwr1	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	DiagTemEnvPwr2	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	DiagTemEnvPwr3	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	DiagTemEnvPwr4	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	DiagTemEnvPwr5	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr0	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr1	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr2	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr3	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr4	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASI	TemEnvPwr5	LHK/tlm/_/B.TKRBIASI.shtml:%	TKR Bias Current ADC value
TKRBIASV	DiagTemEnvPwr0	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	DiagTemEnvPwr1	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	DiagTemEnvPwr2	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	DiagTemEnvPwr3	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	DiagTemEnvPwr4	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	DiagTemEnvPwr5	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	TemEnvPwr0	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value

Bitfield (L)	Tlm. Packet (L)	Section	Description (L)
TKRBIASV	TemEnvPwr1	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	TemEnvPwr2	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	TemEnvPwr3	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	TemEnvPwr4	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRBIASV	TemEnvPwr5	LHK/tlm/_/B.TKRBIASV.shtml:%	TKR Bias Voltage ADC value
TKRCBLT	DiagTemEnvTemp0	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp1	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp2	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp3	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp4	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp5	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp6	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	DiagTemEnvTemp7	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp0	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp1	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp2	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp3	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp4	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp5	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp6	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
TKRCBLT	TemEnvTemp7	LHK/tlm/_/B.TKRCBLT.shtml:%	TKR Cable Temperature ADC va
VCHPDSHPTEMP	DiagPduEnv2	LHK/tlm/_/B.VCHPDSHPTEMP.shtml:%	VCHP-DSHP Interface Temperat
VCHPDSHPTEMP	DiagPduEnv6	LHK/tlm/_/B.VCHPDSHPTEMP.shtml:%	VCHP-DSHP Interface Temperat
VCHPDSHPTEMP	PduEnv2	LHK/tlm/_/B.VCHPDSHPTEMP.shtml:%	VCHP-DSHP Interface Temperat
VCHPDSHPTEMP	PduEnv6	LHK/tlm/_/B.VCHPDSHPTEMP.shtml:%	VCHP-DSHP Interface Temperat
VCHPRSVRHTRTEMP	DiagPduEnv2	LHK/tlm/_/B.VCHPRSVRHTRTEMP.shtml:%	VCHP Reservoir Heater Tempera
VCHPRSVRHTRTEMP	DiagPduEnv6	LHK/tlm/_/B.VCHPRSVRHTRTEMP.shtml:%	VCHP Reservoir Heater Tempera
VCHPRSVRHTRTEMP	PduEnv2	LHK/tlm/_/B.VCHPRSVRHTRTEMP.shtml:%	VCHP Reservoir Heater Tempera
VCHPRSVRHTRTEMP	PduEnv6	LHK/tlm/_/B.VCHPRSVRHTRTEMP.shtml:%	VCHP Reservoir Heater Tempera
VCHPXLHPTEMP	DiagPduEnv2	LHK/tlm/_/B.VCHPXLHPTEMP.shtml:%	VCHP-XLHP Interface Temperat
VCHPXLHPTEMP	DiagPduEnv6	LHK/tlm/_/B.VCHPXLHPTEMP.shtml:%	VCHP-XLHP Interface Temperat
VCHPXLHPTEMP	PduEnv2	LHK/tlm/_/B.VCHPXLHPTEMP.shtml:%	VCHP-XLHP Interface Temperat
VCHPXLHPTEMP	PduEnv6	LHK/tlm/_/B.VCHPXLHPTEMP.shtml:%	VCHP-XLHP Interface Temperat

28 Telemetry Field Index, by Name (LCAT)

Field (L)	Tlm. Packet (L)	Section	Description (L)
ActiveHtPipe	DiagLTC	LTC/tlm/_/F.ActiveHtPipe.shtml:%	Mask for active heat pipes
ADCS	AemEnv0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagAemEnv0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv6	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagPduEnv7	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvPwr5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp6	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	DiagTemEnvTemp7	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv6	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	PduEnv7	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvPwr5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp0	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp1	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp2	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp3	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp4	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp5	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp6	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCS	TemEnvTemp7	LHK/tlm/_/F.ADCS.shtml:%	ADC status bits
ADCV	AemEnv0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagAemEnv0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value

Field (L)	Tlm. Packet (L)	Section	Description (L)
ADCV	DiagPduEnv2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv6	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagPduEnv7	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvPwr5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp6	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	DiagTemEnvTemp7	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv6	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	PduEnv7	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvPwr5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp0	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp1	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp2	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp3	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp4	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp5	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp6	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
ADCV	TemEnvTemp7	LHK/tlm/_/F.ADCV.shtml:%	ADC raw value
AEMFRPWRST	AemEnv0	LHK/tlm/_/F.AEMFRPWRST.shtml:%	AEM FREE Board Power Stat
AEMFRPWRST	DiagAemEnv0	LHK/tlm/_/F.AEMFRPWRST.shtml:%	AEM FREE Board Power Stat
Apid	CmdConfirm	ITC/tlm/_/F.Apid.shtml:%	CCSDS application ID
BOOTDUMPADDR	LBTEPU0HKP	PBC/tlm/_/F.BOOTDUMPADDR.shtml:%	Memory dump address
BOOTDUMPADDR	LBTHKP	PBC/tlm/_/F.BOOTDUMPADDR.shtml:%	Memory dump address
BOOTDUMPCOUNT	LBTEPU0HKP	PBC/tlm/_/F.BOOTDUMPCOUNT.shtml:%	Memory dump word count
BOOTDUMPCOUNT	LBTHKP	PBC/tlm/_/F.BOOTDUMPCOUNT.shtml:%	Memory dump word count
BOOTDUMPDATA	LBTEPU0HKP	PBC/tlm/_/F.BOOTDUMPDATA.shtml:%	Memory dump data
BOOTDUMPDATA	LBTHKP	PBC/tlm/_/F.BOOTDUMPDATA.shtml:%	Memory dump data
BOOTERRWORD	LBTEPU0HKP	PBC/tlm/_/F.BOOTERRWORD.shtml:%	Error code

Field (L)	Tlm. Packet (L)	Section	Description (L)
BOOTERRWORD	LBTHKP	PBC/tlm/_/F.BOOTERRWORD.shtml:%	Error code
BOOTFILEPKT	LBTEPU0HKP	PBC/tlm/_/F.BOOTFILEPKT.shtml:%	File upload packet count
BOOTFILEPKT	LBTHKP	PBC/tlm/_/F.BOOTFILEPKT.shtml:%	File upload packet count
BOOTFILESTATE	LBTEPU0HKP	PBC/tlm/_/F.BOOTFILESTATE.shtml:%	File upload state
BOOTFILESTATE	LBTHKP	PBC/tlm/_/F.BOOTFILESTATE.shtml:%	File upload state
BOOTLASTAPID	LBTEPU0HKP	PBC/tlm/_/F.BOOTLASTAPID.shtml:%	Last APID
BOOTLASTAPID	LBTHKP	PBC/tlm/_/F.BOOTLASTAPID.shtml:%	Last APID
BOOTLASTERR	LBTEPU0HKP	PBC/tlm/_/F.BOOTLASTERR.shtml:%	Most recent boot error code
BOOTLASTERR	LBTHKP	PBC/tlm/_/F.BOOTLASTERR.shtml:%	Most recent boot error code
BOOTLASTFUNC	LBTEPU0HKP	PBC/tlm/_/F.BOOTLASTFUNC.shtml:%	Last Function Code
BOOTLASTFUNC	LBTHKP	PBC/tlm/_/F.BOOTLASTFUNC.shtml:%	Last Function Code
BOOTPKTACCEPT	LBTEPU0HKP	PBC/tlm/_/F.BOOTPKTACCEPT.shtml:%	Accepted packet count
BOOTPKTACCEPT	LBTHKP	PBC/tlm/_/F.BOOTPKTACCEPT.shtml:%	Accepted packet count
BOOTPKTCOUNT	LBTEPU0HKP	PBC/tlm/_/F.BOOTPKTCOUNT.shtml:%	Received packet count
BOOTPKTCOUNT	LBTHKP	PBC/tlm/_/F.BOOTPKTCOUNT.shtml:%	Received packet count
BOOTQERR	LBTEPU0HKP	PBC/tlm/_/F.BOOTQERR.shtml:%	Number of queued errors
BOOTQERR	LBTHKP	PBC/tlm/_/F.BOOTQERR.shtml:%	Number of queued errors
BOOTSCRUBADDRHI	LBTEPU0HKP	PBC/tlm/_/F.BOOTSCRUBADDRHI.shtml:%	Hi 16 bits of software scrub ad
BOOTSCRUBADDRHI	LBTHKP	PBC/tlm/_/F.BOOTSCRUBADDRHI.shtml:%	Hi 16 bits of software scrub ad
BOOTSPARE1	LBTEPU0HKP	PBC/tlm/_/F.BOOTSPARE1.shtml:%	Unused location
BOOTSPARE1	LBTHKP	PBC/tlm/_/F.BOOTSPARE1.shtml:%	Unused location
BOOTSPARE2	LBTEPU0HKP	PBC/tlm/_/F.BOOTSPARE2.shtml:%	Unused location
BOOTSPARE2	LBTHKP	PBC/tlm/_/F.BOOTSPARE2.shtml:%	Unused location
BOOTSWMODE	LBTEPU0HKP	PBC/tlm/_/F.BOOTSWMODE.shtml:%	Current PBC operating mode
BOOTSWMODE	LBTHKP	PBC/tlm/_/F.BOOTSWMODE.shtml:%	Current PBC operating mode
BOOTTOTALERR	LBTEPU0HKP	PBC/tlm/_/F.BOOTTOTALERR.shtml:%	Number of errors encountered
BOOTTOTALERR	LBTHKP	PBC/tlm/_/F.BOOTTOTALERR.shtml:%	Number of errors encountered
BOOTTYPE	LBTEPU0HKP	PBC/tlm/_/F.BOOTTYPE.shtml:%	Boot type
BOOTTYPE	LBTHKP	PBC/tlm/_/F.BOOTTYPE.shtml:%	Boot type
Cmd_len	CmdConfirm	ITC/tlm/_/F.Cmd_len.shtml:%	CCSDS telecommand packet l
CMDCNTS	CmdCnt0	LHK/tlm/_/F.CMDCNTS.shtml:%	Commands sent counter
CMDCNTS	CmdCnt1	LHK/tlm/_/F.CMDCNTS.shtml:%	Commands sent counter
CMDCNTS	DiagCmdCnt0	LHK/tlm/_/F.CMDCNTS.shtml:%	Commands sent counter
CMDCNTS	DiagCmdCnt1	LHK/tlm/_/F.CMDCNTS.shtml:%	Commands sent counter
CMDDISPF	CmdCnt0	LHK/tlm/_/F.CMDDISPF.shtml:%	Command dispatch failure cou
CMDDISPF	CmdCnt1	LHK/tlm/_/F.CMDDISPF.shtml:%	Command dispatch failure cou
CMDDISPF	DiagCmdCnt0	LHK/tlm/_/F.CMDDISPF.shtml:%	Command dispatch failure cou
CMDDISPF	DiagCmdCnt1	LHK/tlm/_/F.CMDDISPF.shtml:%	Command dispatch failure cou
CMDEXEF	CmdCnt0	LHK/tlm/_/F.CMDEXEF.shtml:%	Command execution failure co
CMDEXEF	CmdCnt1	LHK/tlm/_/F.CMDEXEF.shtml:%	Command execution failure co
CMDEXEF	DiagCmdCnt0	LHK/tlm/_/F.CMDEXEF.shtml:%	Command execution failure co
CMDEXEF	DiagCmdCnt1	LHK/tlm/_/F.CMDEXEF.shtml:%	Command execution failure co
counter	acd_cnt	LMC/tlm/_/F.counter.shtml:%	Counter data
counter	cal_cnt	LMC/tlm/_/F.counter.shtml:%	Counter data
counter	tkr_cnt	LMC/tlm/_/F.counter.shtml:%	Counter data
CPUJT	CpuMetr	LHK/tlm/_/F.CPUJT.shtml:%	CPU Junction Temperature
CPUJT	DiagCpuMetr	LHK/tlm/_/F.CPUJT.shtml:%	CPU Junction Temperature
CRXBCNT	CpuMetr	LHK/tlm/_/F.CRXBCNT.shtml:%	1553 Cmd Rx Byte Count
CRXBCNT	DiagCpuMetr	LHK/tlm/_/F.CRXBCNT.shtml:%	1553 Cmd Rx Byte Count
CRXPCNT	CpuMetr	LHK/tlm/_/F.CRXPCNT.shtml:%	1553 Cmd Rx Packet Count
CRXPCNT	DiagCpuMetr	LHK/tlm/_/F.CRXPCNT.shtml:%	1553 Cmd Rx Packet Count
CTXBCNT	CpuMetr	LHK/tlm/_/F.CTXBCNT.shtml:%	1553 Cmd Tx Byte Count

Field (L)	Tlm. Packet (L)	Section	Description (L)
CTXBCNT	DiagCpuMetr	LHK/tlm/_/F.CTXBCNT.shtml:%	1553 Cmd Tx Byte Count
CTXPCNT	CpuMetr	LHK/tlm/_/F.CTXPCNT.shtml:%	1553 Cmd Tx Packet Count
CTXPCNT	DiagCpuMetr	LHK/tlm/_/F.CTXPCNT.shtml:%	1553 Cmd Tx Packet Count
dev_mask	cal_cnt	LMC/tlm/_/F.dev_mask.shtml:%	TEM mask
dev_mask	tkr_cnt	LMC/tlm/_/F.dev_mask.shtml:%	TEM mask
Drop	CmdConfirm	ITC/tlm/_/F.Drop.shtml:%	Number of dropped confirmati
dtime	acd_cnt	LMC/tlm/_/F.dtime.shtml:%	Delta timestamp
dtime	cal_cnt	LMC/tlm/_/F.dtime.shtml:%	Delta timestamp
dtime	tkr_cnt	LMC/tlm/_/F.dtime.shtml:%	Delta timestamp
FILEDATA	LLFSDUMPCTDB	LFS/tlm/_/F.FILEDATA.shtml:%	File Data Contents
FILEDEV	LLFSDIRLIST	LFS/tlm/_/F.FILEDEV.shtml:%	File ID Device Number
FILEDEV	LLFSDUMPCTDB	LFS/tlm/_/F.FILEDEV.shtml:%	File ID Device Number
FILEDEV	LLFSROOTLIST	LFS/tlm/_/F.FILEDEV.shtml:%	File ID Device Number
FILEDEV	LLFSSYSLIST	LFS/tlm/_/F.FILEDEV.shtml:%	File ID Device Number
FILEDIR	LLFSDIRLIST	LFS/tlm/_/F.FILEDIR.shtml:%	File ID Directory Number
FILEDIR	LLFSDUMPCTDB	LFS/tlm/_/F.FILEDIR.shtml:%	File ID Directory Number
FILEDIR	LLFSROOTLIST	LFS/tlm/_/F.FILEDIR.shtml:%	File ID Directory Number
FILEDIR	LLFSSYSLIST	LFS/tlm/_/F.FILEDIR.shtml:%	File ID Directory Number
FILEDUMPSIZE	LLFSDUMPCTDB	LFS/tlm/_/F.FILEDUMPSIZE.shtml:%	File Dump Current Size
FILEHDR	LLFSDIRLIST	LFS/tlm/_/F.FILEHDR.shtml:%	File Header Data
FILEIDCOM	DiagFileStats	LHK/tlm/_/F.FILEIDCOM.shtml:%	FILE ID Commit
FILEIDCOM	FileStats	LHK/tlm/_/F.FILEIDCOM.shtml:%	FILE ID Commit
FILENUM	LLFSDIRLIST	LFS/tlm/_/F.FILENUM.shtml:%	File ID File Number
FILENUM	LLFSDUMPCTDB	LFS/tlm/_/F.FILENUM.shtml:%	File ID File Number
FILENUM	LLFSROOTLIST	LFS/tlm/_/F.FILENUM.shtml:%	File ID File Number
FILENUM	LLFSSYSLIST	LFS/tlm/_/F.FILENUM.shtml:%	File ID File Number
FILEPKTCNT	DiagFileStats	LHK/tlm/_/F.FILEPKTCNT.shtml:%	FILE Packet Count
FILEPKTCNT	FileStats	LHK/tlm/_/F.FILEPKTCNT.shtml:%	FILE Packet Count
FILERRCNT	DiagFileStats	LHK/tlm/_/F.FILERRCNT.shtml:%	FILE Error Count
FILERRCNT	FileStats	LHK/tlm/_/F.FILERRCNT.shtml:%	FILE Error Count
FILERRCODE	DiagFileStats	LHK/tlm/_/F.FILERRCODE.shtml:%	FILE Error Code
FILERRCODE	FileStats	LHK/tlm/_/F.FILERRCODE.shtml:%	FILE Error Code
FILESIZECUR	DiagFileStats	LHK/tlm/_/F.FILESIZECUR.shtml:%	File Size Current
FILESIZECUR	FileStats	LHK/tlm/_/F.FILESIZECUR.shtml:%	File Size Current
FILESTARCHIVE	LLFSDIRLIST	LFS/tlm/_/F.FILESTARCHIVE.shtml:%	File Storage Archive Flag
FILESTARCHIVE	LLFSROOTLIST	LFS/tlm/_/F.FILESTARCHIVE.shtml:%	File Storage Archive Flag
FILESTATE	DiagFileStats	LHK/tlm/_/F.FILESTATE.shtml:%	FILE Upload State
FILESTATE	FileStats	LHK/tlm/_/F.FILESTATE.shtml:%	FILE Upload State
FILESTBLOCK	LLFSDIRLIST	LFS/tlm/_/F.FILESTBLOCK.shtml:%	File Storage Blocks
FILESTBLOCK	LLFSROOTLIST	LFS/tlm/_/F.FILESTBLOCK.shtml:%	File Storage Blocks
FILESTDIR	LLFSDIRLIST	LFS/tlm/_/F.FILESTDIR.shtml:%	File Storage Directory Flag
FILESTDIR	LLFSROOTLIST	LFS/tlm/_/F.FILESTDIR.shtml:%	File Storage Directory Flag
FILESTOFFSET	LLFSDUMPCTDB	LFS/tlm/_/F.FILESTOFFSET.shtml:%	File Data Offset
FILESTRDONLY	LLFSDIRLIST	LFS/tlm/_/F.FILESTRDONLY.shtml:%	File Storage Read-Only Flag
FILESTRDONLY	LLFSROOTLIST	LFS/tlm/_/F.FILESTRDONLY.shtml:%	File Storage Read-Only Flag
FILESTSIZE	LLFSDIRLIST	LFS/tlm/_/F.FILESTSIZE.shtml:%	File Storage Size
FILESTSIZE	LLFSROOTLIST	LFS/tlm/_/F.FILESTSIZE.shtml:%	File Storage Size
FILESTTIME	LLFSDIRLIST	LFS/tlm/_/F.FILESTTIME.shtml:%	File Storage Update Time
FILESTTIME	LLFSROOTLIST	LFS/tlm/_/F.FILESTTIME.shtml:%	File Storage Update Time
Fnc_code	CmdConfirm	ITC/tlm/_/F.Fnc_code.shtml:%	CCSDS telecommand functio
Fnc_pad	CmdConfirm	ITC/tlm/_/F.Fnc_pad.shtml:%	Padding
GEMLRSDISC	DiagLrs0	LHK/tlm/_/F.GEMLRSDISC.shtml:%	GEM low-rate science discard

Field (L)	Tlm. Packet (L)	Section	Description (L)
GEMLRSDISC	Lrs0	LHK/tlm/_/F.GEMLRSDISC.shtml:%	GEM low-rate science discard
GEMLRSLIVE	DiagLrs0	LHK/tlm/_/F.GEMLRSLIVE.shtml:%	GEM low-rate science livetim
GEMLRSLIVE	Lrs0	LHK/tlm/_/F.GEMLRSLIVE.shtml:%	GEM low-rate science livetim
GEMLRSSENT	DiagLrs0	LHK/tlm/_/F.GEMLRSSENT.shtml:%	GEM low-rate science sent co
GEMLRSSENT	Lrs0	LHK/tlm/_/F.GEMLRSSENT.shtml:%	GEM low-rate science sent co
GEMPRSCL	DiagLrs0	LHK/tlm/_/F.GEMPRSCL.shtml:%	Low-rate science prescaled co
GEMPRSCL	Lrs0	LHK/tlm/_/F.GEMPRSCL.shtml:%	Low-rate science prescaled co
HeaterCmdMask	DiagLTC	LTC/tlm/_/F.HeaterCmdMask.shtml:%	Command mask for reservoir I
HKBCNT	CpuMetr	LHK/tlm/_/F.HKBCNT.shtml:%	1553 Housekeeping Byte Cou
HKBCNT	DiagCpuMetr	LHK/tlm/_/F.HKBCNT.shtml:%	1553 Housekeeping Byte Cou
HKPCNT	CpuMetr	LHK/tlm/_/F.HKPCNT.shtml:%	1553 Housekeeping Packet Co
HKPCNT	DiagCpuMetr	LHK/tlm/_/F.HKPCNT.shtml:%	1553 Housekeeping Packet Co
INTRCNT	CpuMetr	LHK/tlm/_/F.INTRCNT.shtml:%	1553 Interrupts
INTRCNT	DiagCpuMetr	LHK/tlm/_/F.INTRCNT.shtml:%	1553 Interrupts
isCmd	CmdConfirm	ITC/tlm/_/F.isCmd.shtml:%	CCSDS command bit
ITC_NodeID	CmdConfirm	ITC/tlm/_/F.ITC_NodeID.shtml:%	ITC node ID
ITC_TaskID	CmdConfirm	ITC/tlm/_/F.ITC_TaskID.shtml:%	ITC task ID
LATSTUNIT	LLFSDIRLIST	LFS/tlm/_/F.LATSTUNIT.shtml:%	LAT Storage Unit
LATSTUNIT	LLFSDUMPCTDB	LFS/tlm/_/F.LATSTUNIT.shtml:%	LAT Storage Unit
LATSTUNIT	LLFSROOTLIST	LFS/tlm/_/F.LATSTUNIT.shtml:%	LAT Storage Unit
LATSTUNIT	LLFSSYSLIST	LFS/tlm/_/F.LATSTUNIT.shtml:%	LAT Storage Unit
LFSXID	LLFSDIRLIST	LFS/tlm/_/F.LFSXID.shtml:%	Transaction ID
LFSXID	LLFSDUMPCTDB	LFS/tlm/_/F.LFSXID.shtml:%	Transaction ID
LFSXID	LLFSROOTLIST	LFS/tlm/_/F.LFSXID.shtml:%	Transaction ID
LFSXID	LLFSSYSLIST	LFS/tlm/_/F.LFSXID.shtml:%	Transaction ID
lrs_mask	cal_cnt	LMC/tlm/_/F.lrs_mask.shtml:%	Low Rate Science Mask
lrs_mask	tkr_cnt	LMC/tlm/_/F.lrs_mask.shtml:%	Low Rate Science Mask
MDPACT	DiagMemStats0	LHK/tlm/_/F.MDPACT.shtml:%	Memory dump active
MDPACT	DiagMemStats1	LHK/tlm/_/F.MDPACT.shtml:%	Memory dump active
MDPACT	MemStats0	LHK/tlm/_/F.MDPACT.shtml:%	Memory dump active
MDPACT	MemStats1	LHK/tlm/_/F.MDPACT.shtml:%	Memory dump active
MDPADDR	DiagMemStats0	LHK/tlm/_/F.MDPADDR.shtml:%	Memory dump address
MDPADDR	DiagMemStats1	LHK/tlm/_/F.MDPADDR.shtml:%	Memory dump address
MDPADDR	MemStats0	LHK/tlm/_/F.MDPADDR.shtml:%	Memory dump address
MDPADDR	MemStats1	LHK/tlm/_/F.MDPADDR.shtml:%	Memory dump address
MDPBYTS	DiagMemStats0	LHK/tlm/_/F.MDPBYTS.shtml:%	Memory dump bytes
MDPBYTS	DiagMemStats1	LHK/tlm/_/F.MDPBYTS.shtml:%	Memory dump bytes
MDPBYTS	MemStats0	LHK/tlm/_/F.MDPBYTS.shtml:%	Memory dump bytes
MDPBYTS	MemStats1	LHK/tlm/_/F.MDPBYTS.shtml:%	Memory dump bytes
MDPFCD	DiagMemStats0	LHK/tlm/_/F.MDPFCD.shtml:%	Memory dump function code
MDPFCD	DiagMemStats1	LHK/tlm/_/F.MDPFCD.shtml:%	Memory dump function code
MDPFCD	MemStats0	LHK/tlm/_/F.MDPFCD.shtml:%	Memory dump function code
MDPFCD	MemStats1	LHK/tlm/_/F.MDPFCD.shtml:%	Memory dump function code
MDPSTADR	DiagMemStats0	LHK/tlm/_/F.MDPSTADR.shtml:%	Memory dump start address
MDPSTADR	DiagMemStats1	LHK/tlm/_/F.MDPSTADR.shtml:%	Memory dump start address
MDPSTADR	MemStats0	LHK/tlm/_/F.MDPSTADR.shtml:%	Memory dump start address
MDPSTADR	MemStats1	LHK/tlm/_/F.MDPSTADR.shtml:%	Memory dump start address
MDPSTAT	DiagMemStats0	LHK/tlm/_/F.MDPSTAT.shtml:%	Memory dump status
MDPSTAT	DiagMemStats1	LHK/tlm/_/F.MDPSTAT.shtml:%	Memory dump status
MDPSTAT	MemStats0	LHK/tlm/_/F.MDPSTAT.shtml:%	Memory dump status
MDPSTAT	MemStats1	LHK/tlm/_/F.MDPSTAT.shtml:%	Memory dump status
MDPTXID	DiagMemStats0	LHK/tlm/_/F.MDPTXID.shtml:%	Memory dump transaction ID

Field (L)	Tlm. Packet (L)	Section	Description (L)
MDPTXID	DiagMemStats1	LHK/tlm/_/F.MDPTXID.shtml:%	Memory dump transaction ID
MDPTXID	MemStats0	LHK/tlm/_/F.MDPTXID.shtml:%	Memory dump transaction ID
MDPTXID	MemStats1	LHK/tlm/_/F.MDPTXID.shtml:%	Memory dump transaction ID
MLDACT	DiagMemStats0	LHK/tlm/_/F.MLDACT.shtml:%	Memory load active flag
MLDACT	DiagMemStats1	LHK/tlm/_/F.MLDACT.shtml:%	Memory load active flag
MLDACT	MemStats0	LHK/tlm/_/F.MLDACT.shtml:%	Memory load active flag
MLDACT	MemStats1	LHK/tlm/_/F.MLDACT.shtml:%	Memory load active flag
MLDBYTS	DiagMemStats0	LHK/tlm/_/F.MLDBYTS.shtml:%	Memory load total bytes
MLDBYTS	DiagMemStats1	LHK/tlm/_/F.MLDBYTS.shtml:%	Memory load total bytes
MLDBYTS	MemStats0	LHK/tlm/_/F.MLDBYTS.shtml:%	Memory load total bytes
MLDBYTS	MemStats1	LHK/tlm/_/F.MLDBYTS.shtml:%	Memory load total bytes
MLDOFF	DiagMemStats0	LHK/tlm/_/F.MLDOFF.shtml:%	Memory load offset
MLDOFF	DiagMemStats1	LHK/tlm/_/F.MLDOFF.shtml:%	Memory load offset
MLDOFF	MemStats0	LHK/tlm/_/F.MLDOFF.shtml:%	Memory load offset
MLDOFF	MemStats1	LHK/tlm/_/F.MLDOFF.shtml:%	Memory load offset
MLDSTADR	DiagMemStats0	LHK/tlm/_/F.MLDSTADR.shtml:%	Starting memory load address
MLDSTADR	DiagMemStats1	LHK/tlm/_/F.MLDSTADR.shtml:%	Starting memory load address
MLDSTADR	MemStats0	LHK/tlm/_/F.MLDSTADR.shtml:%	Starting memory load address
MLDSTADR	MemStats1	LHK/tlm/_/F.MLDSTADR.shtml:%	Starting memory load address
MLDSTAT	DiagMemStats0	LHK/tlm/_/F.MLDSTAT.shtml:%	Status of most recent load acti
MLDSTAT	DiagMemStats1	LHK/tlm/_/F.MLDSTAT.shtml:%	Status of most recent load acti
MLDSTAT	MemStats0	LHK/tlm/_/F.MLDSTAT.shtml:%	Status of most recent load acti
MLDSTAT	MemStats1	LHK/tlm/_/F.MLDSTAT.shtml:%	Status of most recent load acti
Pad2	CmdConfirm	ITC/tlm/_/F.Pad2.shtml:%	Pad two bytes
PDUACDCNVT	DiagPduEnv0	LHK/tlm/_/F.PDUACDCNVT.shtml:%	PDU ACD Power Converter S
PDUACDCNVT	DiagPduEnv4	LHK/tlm/_/F.PDUACDCNVT.shtml:%	PDU ACD Power Converter S
PDUACDCNVT	PduEnv0	LHK/tlm/_/F.PDUACDCNVT.shtml:%	PDU ACD Power Converter S
PDUACDCNVT	PduEnv4	LHK/tlm/_/F.PDUACDCNVT.shtml:%	PDU ACD Power Converter S
PDUACDPWRST	DiagPduEnv0	LHK/tlm/_/F.PDUACDPWRST.shtml:%	PDU ACD Power State
PDUACDPWRST	DiagPduEnv4	LHK/tlm/_/F.PDUACDPWRST.shtml:%	PDU ACD Power State
PDUACDPWRST	PduEnv0	LHK/tlm/_/F.PDUACDPWRST.shtml:%	PDU ACD Power State
PDUACDPWRST	PduEnv4	LHK/tlm/_/F.PDUACDPWRST.shtml:%	PDU ACD Power State
PDUACDPWRSUP	DiagPduEnv0	LHK/tlm/_/F.PDUACDPWRSUP.shtml:%	PDU ACD Power Supply Sour
PDUACDPWRSUP	DiagPduEnv4	LHK/tlm/_/F.PDUACDPWRSUP.shtml:%	PDU ACD Power Supply Sour
PDUACDPWRSUP	PduEnv0	LHK/tlm/_/F.PDUACDPWRSUP.shtml:%	PDU ACD Power Supply Sour
PDUACDPWRSUP	PduEnv4	LHK/tlm/_/F.PDUACDPWRSUP.shtml:%	PDU ACD Power Supply Sour
PDUEPUCVTST	DiagPduEnv0	LHK/tlm/_/F.PDUEPUCVTST.shtml:%	PDU EPU Converter State
PDUEPUCVTST	DiagPduEnv4	LHK/tlm/_/F.PDUEPUCVTST.shtml:%	PDU EPU Converter State
PDUEPUCVTST	PduEnv0	LHK/tlm/_/F.PDUEPUCVTST.shtml:%	PDU EPU Converter State
PDUEPUCVTST	PduEnv4	LHK/tlm/_/F.PDUEPUCVTST.shtml:%	PDU EPU Converter State
PDUEPUPWRST	DiagPduEnv0	LHK/tlm/_/F.PDUEPUPWRST.shtml:%	PDU EPU Crate Power State
PDUEPUPWRST	DiagPduEnv4	LHK/tlm/_/F.PDUEPUPWRST.shtml:%	PDU EPU Crate Power State
PDUEPUPWRST	PduEnv0	LHK/tlm/_/F.PDUEPUPWRST.shtml:%	PDU EPU Crate Power State
PDUEPUPWRST	PduEnv4	LHK/tlm/_/F.PDUEPUPWRST.shtml:%	PDU EPU Crate Power State
PDUTEMPWRST	DiagPduEnv0	LHK/tlm/_/F.PDUTEMPWRST.shtml:%	PDU TEM Power State
PDUTEMPWRST	DiagPduEnv4	LHK/tlm/_/F.PDUTEMPWRST.shtml:%	PDU TEM Power State
PDUTEMPWRST	PduEnv0	LHK/tlm/_/F.PDUTEMPWRST.shtml:%	PDU TEM Power State
PDUTEMPWRST	PduEnv4	LHK/tlm/_/F.PDUTEMPWRST.shtml:%	PDU TEM Power State
RawAdcStatVal	DiagLTC	LTC/tlm/_/F.RawAdcStatVal.shtml:%	Raw ADC Status and Value
RLADC	RedLimAlrt	LHK/tlm/_/F.RLADC.shtml:%	Red Limit ADC Number
RLCNT	RedLimAlrt	LHK/tlm/_/F.RLCNT.shtml:%	Red Limit ADC Counts
RLDEV	RedLimAlrt	LHK/tlm/_/F.RLDEV.shtml:%	Red Limit Device Opcode

Field (L)	Tlm. Packet (L)	Section	Description (L)
RLLIM	RedLimAlrt	LHK/tlm/_/F.RLLIM.shtml:%	Red Limit Threshold
RTERR	CpuMetr	LHK/tlm/_/F.RTERR.shtml:%	1553 Error Count
RTERR	DiagCpuMetr	LHK/tlm/_/F.RTERR.shtml:%	1553 Error Count
Sec_hdr	CmdConfirm	ITC/tlm/_/F.Sec_hdr.shtml:%	CCSDS secondary header flag
SensorHpNum	DiagLTC	LTC/tlm/_/F.SensorHpNum.shtml:%	Specifies connected HP numbe
SensorStatus	DiagLTC	LTC/tlm/_/F.SensorStatus.shtml:%	Status for HP sensors
SensorType	DiagLTC	LTC/tlm/_/F.SensorType.shtml:%	Specifies Sensor Type
Seq_cnt	CmdConfirm	ITC/tlm/_/F.Seq_cnt.shtml:%	CCSDS sequencing count
Seq_flg	CmdConfirm	ITC/tlm/_/F.Seq_flg.shtml:%	CCSDS sequencing bits
spare16	acd_cnt	LMC/tlm/_/F.spare16.shtml:%	Spare 16 bits
Status	CmdConfirm	ITC/tlm/_/F.Status.shtml:%	MSG status code
SYSBLKFREE	LLFSSYSLIST	LFS/tlm/_/F.SYSBLKFREE.shtml:%	File System Free Blocks
SYSBLKSIZE	LLFSSYSLIST	LFS/tlm/_/F.SYSBLKSIZE.shtml:%	File System Block Size
SYSBLKTOTAL	LLFSSYSLIST	LFS/tlm/_/F.SYSBLKTOTAL.shtml:%	File System Total Blocks
TDUMPADDR	LMEMSIUDATA	MEM/tlm/_/F.TDUMPADDR.shtml:%	Starting Address
TDUMPCMDFUNC	LMEMSIUDATA	MEM/tlm/_/F.TDUMPCMDFUNC.shtml:%	Dump Type
TDUMPDATA	LMEMSIUDATA	MEM/tlm/_/F.TDUMPDATA.shtml:%	Dump Data Word
TDUMPSIZE	LMEMSIUDATA	MEM/tlm/_/F.TDUMPSIZE.shtml:%	Word Count
TEMDEADTIME	DiagLrs0	LHK/tlm/_/F.TEMDEADTIME.shtml:%	TEM Deadtime Low-rate Scie
TEMDEADTIME	Lrs0	LHK/tlm/_/F.TEMDEADTIME.shtml:%	TEM Deadtime Low-rate Scie
TempCelsius	DiagLTC	LTC/tlm/_/F.TempCelsius.shtml:%	Temperature converted to celsi
tile_id	acd_cnt	LMC/tlm/_/F.tile_id.shtml:%	ACD Tile ID
Time_lsui	CmdConfirm	ITC/tlm/_/F.Time_lsui.shtml:%	Time (least significant 32 bits)
Time_msui	CmdConfirm	ITC/tlm/_/F.Time_msui.shtml:%	Time (most significant 32 bits)
TIMESEC	CmdCnt0	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESEC	CmdCnt1	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESEC	DiagCmdCnt0	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESEC	DiagCmdCnt1	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESEC	DiagLrs0	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESEC	Lrs0	LHK/tlm/_/F.TIMESEC.shtml:%	Timestamp seconds
TIMESUBSEC	CmdCnt0	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TIMESUBSEC	CmdCnt1	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TIMESUBSEC	DiagCmdCnt0	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TIMESUBSEC	DiagCmdCnt1	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TIMESUBSEC	DiagLrs0	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TIMESUBSEC	Lrs0	LHK/tlm/_/F.TIMESUBSEC.shtml:%	Timestamp subseconds
TLATUNIT	LMEMPOOLDATA	MEM/tlm/_/F.TLATUNIT.shtml:%	Source LAT Unit
TLATUNIT	LMEMSIUDATA	MEM/tlm/_/F.TLATUNIT.shtml:%	Source LAT Unit
TLATUNIT	LMEMSYMVAL	MEM/tlm/_/F.TLATUNIT.shtml:%	Source LAT Unit
TLMBCNT	CpuMetr	LHK/tlm/_/F.TLMBCNT.shtml:%	1553 Telemertry Byte Count
TLMBCNT	DiagCpuMetr	LHK/tlm/_/F.TLMBCNT.shtml:%	1553 Telemertry Byte Count
TLMPCNT	CpuMetr	LHK/tlm/_/F.TLMPCNT.shtml:%	1553 Telemetry Packet Count
TLMPCNT	DiagCpuMetr	LHK/tlm/_/F.TLMPCNT.shtml:%	1553 Telemetry Packet Count
TPAD16	LMEMPOOLDATA	MEM/tlm/_/F.TPAD16.shtml:%	16-Bit Padding
TPAD16	LMEMSYMVAL	MEM/tlm/_/F.TPAD16.shtml:%	16-Bit Padding
TPAD8	LMEMSYMVAL	MEM/tlm/_/F.TPAD8.shtml:%	8-Bit Padding
TPOOLALLOCBLOCKS	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLALLOCBLOCKS.shtml:%	Memory Pool Allocated Block
TPOOLALLOCBYTES	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLALLOCBYTES.shtml:%	Memory Pool Allocated Bytes
TPOOLFREEBLOCKS	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLFREEBLOCKS.shtml:%	Memory Pool Free Blocks
TPOOLFREEBYTES	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLFREEBYTES.shtml:%	Memory Pool Free Bytes
TPOOLID	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLID.shtml:%	Memory Pool ID
TPOOLMAXBLKBYTES	LMEMPOOLDATA	MEM/tlm/_/F.TPOOLMAXBLKBYTES.shtml:%	Memory Pool Maximum Bloc

Field (L)	Tlm. Packet (L)	Section	Description (L)
TSP10	DiagPduEnv0	LHK/tlm/_/F.TSP10.shtml:%	Spare 10 bits
TSP10	DiagPduEnv4	LHK/tlm/_/F.TSP10.shtml:%	Spare 10 bits
TSP10	PduEnv0	LHK/tlm/_/F.TSP10.shtml:%	Spare 10 bits
TSP10	PduEnv4	LHK/tlm/_/F.TSP10.shtml:%	Spare 10 bits
TSP13	DiagPduEnv0	LHK/tlm/_/F.TSP13.shtml:%	Spare 13 bit field
TSP13	DiagPduEnv4	LHK/tlm/_/F.TSP13.shtml:%	Spare 13 bit field
TSP13	PduEnv0	LHK/tlm/_/F.TSP13.shtml:%	Spare 13 bit field
TSP13	PduEnv4	LHK/tlm/_/F.TSP13.shtml:%	Spare 13 bit field
TSP16	CmdCnt0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	CmdCnt1	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	CpuMetr	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagCmdCnt0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagCmdCnt1	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagCpuMetr	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagFileStats	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagLrs0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagMemStats0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagMemStats1	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	DiagTemEnvPwr5	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	FileStats	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	Lrs0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	MemStats0	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	MemStats1	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP16	TemEnvPwr5	LHK/tlm/_/F.TSP16.shtml:%	Spare 16 bit field
TSP4	AemEnv0	LHK/tlm/_/F.TSP4.shtml:%	Spare 4 bits
TSP4	DiagAemEnv0	LHK/tlm/_/F.TSP4.shtml:%	Spare 4 bits
TSP8	AemEnv0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagAemEnv0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv6	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagPduEnv7	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvPwr5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp6	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	DiagTemEnvTemp7	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field

Field (L)	Tlm. Packet (L)	Section	Description (L)
TSP8	PduEnv2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv6	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	PduEnv7	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvPwr5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp0	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp1	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp2	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp3	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp4	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp5	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp6	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSP8	TemEnvTemp7	LHK/tlm/_/F.TSP8.shtml:%	Spare byte field
TSYMNAMECHAR	LMEMSYMVAL	MEM/tlm/_/F.TSYMNAMECHAR.shtml:%	Symbol Name Character
TSYMNAMESIZE	LMEMSYMVAL	MEM/tlm/_/F.TSYMNAMESIZE.shtml:%	Symbol Name Length
TSYMVAL	LMEMSYMVAL	MEM/tlm/_/F.TSYMVAL.shtml:%	Symbol Value
TTRANID	LMEMPOOLDATA	MEM/tlm/_/F.TTRANID.shtml:%	Transaction ID
TTRANID	LMEMSIUDATA	MEM/tlm/_/F.TTRANID.shtml:%	Transaction ID
TTRANID	LMEMSYMVAL	MEM/tlm/_/F.TTRANID.shtml:%	Transaction ID
Version	CmdConfirm	ITC/tlm/_/F.Version.shtml:%	CCSDS version

29 Telemetry Analog Conv. Index, by Name

30 Telemetry Discrete Conv. Index, by Name

Discrete Conv.	Section	Description
ITC_NODEID	ITC/att/_/D.ITC_NODEID.shtml:%	Discrete list of ITC nodes
ITC_TASKID	ITC/att/_/D.ITC_TASKID.shtml:%	Discrete list of ITC task IDs
LAEMFRPWRSTATES	LHK/att/_/D.LAEMFRPWRSTATES.shtml:%	AEM FREE Board Power States
LAPDUPWRCNVTSTAT	LHK/att/_/D.LAPDUPWRCNVTSTAT.shtml:%	ACD PDU Power Converter State
LAPDUPWRSTATES	LHK/att/_/D.LAPDUPWRSTATES.shtml:%	ACD PDU Power States
LAPDUPWRSUPSTAT	LHK/att/_/D.LAPDUPWRSUPSTAT.shtml:%	ACD PDU Power Supply Sources
LDPDUEPUCNVT	LHK/att/_/D.LDPDUEPUCNVT.shtml:%	PDU EPU Converter
LDPDUEPUPWRST	LHK/att/_/D.LDPDUEPUPWRST.shtml:%	EPU Power States
LDPDUTEMPWRST	LHK/att/_/D.LDPDUTEMPWRST.shtml:%	PDU TEM Power State
LHKSTATUSBITS	LHK/att/_/D.LHKSTATUSBITS.shtml:%	ADC status bit conversions
LRLIMDEVICE	LHK/att/_/D.LRLIMDEVICE.shtml:%	Device Opcodes for Red Limit Alerts

31 Telemetry Limit Set Index, by Name

Limit Set	Section	Description
LABEAGTEMPADCLIM	LHK/att/_/L.LABEAGTEMPADCLIM.shtml:%	ACD BEA Grid Interface Temperature Limits
LAPMTRTEMPADCLIM	LHK/att/_/L.LAPMTRTEMPADCLIM.shtml:%	ACD PMT Rail Temperature Limits
LASHLTEMPADCLIM	LHK/att/_/L.LASHLTEMPADCLIM.shtml:%	ACD Shell Temperature Limits
LC33IADCLIM	LHK/att/_/L.LC33IADCLIM.shtml:%	CAL 3.3I ADC limits
LC33VADCLIM	LHK/att/_/L.LC33VADCLIM.shtml:%	CAL 3.3V ADC limits
LCAFETADCLIM	LHK/att/_/L.LCAFETADCLIM.shtml:%	CAL AFFE temperature limits
LCBASPLADCLIM	LHK/att/_/L.LCBASPLADCLIM.shtml:%	CAL Baseplate Temperature Limits
LCBIASIADCLIM	LHK/att/_/L.LCBIASIADCLIM.shtml:%	CAL bias current limits
LCBIASVADCLIM	LHK/att/_/L.LCBIASVADCLIM.shtml:%	CAL bias voltage limits
LDAEMFRHV1ADCLIM	LHK/att/_/L.LDAEMFRHV1ADCLIM.shtml:%	AEM Free Board HV1 Limits
LDAEMFRHV2ADCLIM	LHK/att/_/L.LDAEMFRHV2ADCLIM.shtml:%	AEM Free Board HV2 Limits
LDAEMFRTMPADCLIM	LHK/att/_/L.LDAEMFRTMPADCLIM.shtml:%	AEM Free Board Temperature Limits
LDAEMFRVDDADCLIM	LHK/att/_/L.LDAEMFRVDDADCLIM.shtml:%	AEM Free Board VDD Limits
LDEPUTEMPADCLIM	LHK/att/_/L.LDEPUTEMPADCLIM.shtml:%	EPU Temperature ADC Limits
LDEPUVADCLIM	LHK/att/_/L.LDEPUVADCLIM.shtml:%	EPU Voltage ADC Limits
LDTEM33IADCLIM	LHK/att/_/L.LDTEM33IADCLIM.shtml:%	TEM 3.3I digital limits
LDTEM33VADCLIM	LHK/att/_/L.LDTEM33VADCLIM.shtml:%	TEM digital 3.3V limits
LDTEMPCBTADCLIM	LHK/att/_/L.LDTEMPCBTADCLIM.shtml:%	TEM PCB Temperature Limits
LDTEMPSTADCLIM	LHK/att/_/L.LDTEMPSTADCLIM.shtml:%	TEM Power Supply Temperature Limits
LMGRDRADIFADCLIM	LHK/att/_/L.LMGRDRADIFADCLIM.shtml:%	Grid Radiator Interface Temperature Limits
LMGRIDTEMPADCLIM	LHK/att/_/L.LMGRIDTEMPADCLIM.shtml:%	Grid Temperature ADC Limits
LMRADAFHTRADCLIM	LHK/att/_/L.LMRADAFHTRADCLIM.shtml:%	Radiator Anitfreeze Heater Temperature Limits
LMRADTEMPADCLIM	LHK/att/_/L.LMRADTEMPADCLIM.shtml:%	Radiator Temperature Limits
LMVCHPDSHPADCLIM	LHK/att/_/L.LMVCHPDSHPADCLIM.shtml:%	VCHP-DSHP Interface Temperature Limits
LMVCHPRSVTADCLIM	LHK/att/_/L.LMVCHPRSVTADCLIM.shtml:%	VCHP Reservoir Heater Temperature Limits
LMVCHPXLHPADCLIM	LHK/att/_/L.LMVCHPXLHPADCLIM.shtml:%	VCHP-XLHP Interface Temperature Limits
LT15IADCLIM	LHK/att/_/L.LT15IADCLIM.shtml:%	TKR 1.5I ADC limits
LT15VADCLIM	LHK/att/_/L.LT15VADCLIM.shtml:%	TKR 1.5V ADC Limits
LT25IADCLIM	LHK/att/_/L.LT25IADCLIM.shtml:%	TKR 2.5I ADC limits
LT25VADCLIM	LHK/att/_/L.LT25VADCLIM.shtml:%	TKR 2.5V ADC limits
LTBIASIADCLIM	LHK/att/_/L.LTBIASIADCLIM.shtml:%	TKR bias current ADC limits
LTBIASVADCLIM	LHK/att/_/L.LTBIASVADCLIM.shtml:%	TKR bias voltage ADC limits
LTCBLTADCLIM	LHK/att/_/L.LTCBLTADCLIM.shtml:%	TKR cable temperature limits