SMEI Instrument Telemetry Format Specification

M.P.Cooke

and

C.J.Eyles

University of Birmingham

Issue 3

21 August 2000

Table of Contents

1. Introduction

The SMEI instrument routinely accumulates science and housekeeping data while it is operating. This document defines the formats and protocols used when transferring this data to the spacecraft.

This document does not cover instrument commanding, which is detailed in the 'SMEI Instrument Commanding Protocol' (SMEI/BU/SPE/001), and the 'SMEI Instrument Commanding Specification' (SMEI/BU/SPE/002).

1 1.1 Nomenclature

The general format used for presenting data is a sequence of 16−bit words with a mask indicating which bits are valid and invalid for each parameter. The most significant bit is shown at the left, the least significant bit at the right of the mask field.

Where a parameter is shown as a field with a number of options, unspecified options are not permitted.

Examples:

Word	Mask	Parameter
0	`xxxxxxxxxxxxxxxx`	All bits are valid.
1	`xxxxxxxx--------`	Bits 15 −8 are valid.
2	`--------xxxxxxxx`	Bits 7 −0 are valid.
3	`--------x-------`	Bit 7 is valid. Eg, 0=Disabled, 1=Enabled.
4	`--------xxxxxxxx`	8 LSBs of parameter A
5	`-----------xxx--`	Bits 4−2 are valid and the meaning is dependant on the sub−mask
	`-----------000--`	option 0
	`-----------001--`	option 1
	`-----------010--`	option 2
6	`---------------1`	Bit 0 is always 1
7−31		These words are not applicable, or are described elsewhere.

2. Data Sources

Science image data and telemetry are routinely accumulated into blocks and stored in the main memory of the data handling unit (DHU). During standard operations, these blocks of data are transmitted to the spacecraft continuously over the MIL−STD−1553B bus. There is no provision onboard SMEI for long term storage of more than a few images.

2.1 State of Health Information

Instrument housekeeping data is grouped into 32−word packets as it is collected. As SMEI produces more than 32 words of housekeeping data, each packet has a type identifier, so we can de−multiplex the data during ground processing. Each state of health packet is time−stamped using the least significant seconds field received from the spacecraft.

2.2 Camera Image Information

SMEI images are (optionally) compressed and error correction encoded by the DHU. Each image is broken into a number of 256−word units of (compressed) data, though the final unit of data for an image is usually smaller than this. Eight words of error correction codes are prefixed to each unit to make an image data packet.

2.3 Spacecraft Time and Attitude Information

The spacecraft provides periodic updates of the current time and the attitude data for the payload. The interface is defined in section C6.2.1 of the Interface Control Document (ICD).

3. SMEI to Spacecraft Data Streams

The SMEI DHU is designed to produce two logical streams of data for the spacecraft. The first is a State of Health (SoH) stream, of 2560 bits per second. This is simply composed of instrument housekeeping packets.

The second stream is the Science Data Stream (SDS), of 64,000 bits per second in normal operating mode, and 128,000 bits per second in engineering mode. This stream is composed of both camera image data, and instrument housekeeping. Appropriate synchronisation header information is included in the data stream to separate the two.

3.1 State Of Health Stream

The generic format of the SoH packet is shown below. It uses a 3 word fixed format header, containing a housekeeping identifier, checksum and timestamp.

The type identifier uses a single bit to identify the type of housekeeping contained in the packet. This leaves four spare identifiers for later additions.

The cyclic redundancy check is generated using the same scheme as WindSat. The details can be found in Appendix C.

The timestamp field is just a copy of the least significant word of the spacecraft time seconds data (SCT Seconds LSW) that the spacecraft supplies to SMEI periodically. This allows good knowledge of when the housekeeping was last updated. Spacecraft time is currently defined as time since noon 1/1/2000 UTC, though this is still to be confirmed.

Word	Mask	Mnemonic	Parameter
0	xxxxxxxxxxxxxxxx	SOH_TYPE	Multiplex Packet Type Identifier
	0000000000000001		Reserved Block
	0000000000000010		Reserved Block
	0000000000000011		Reserved Block
	0000000000000100		Camera 1 Observation Parameters
	0000000000000101		Camera 2 Observation Parameters
	0000000000000110		Camera 3 Observation Parameters
	0000000000000111		Analogue and Digital Monitors
	0000000000001000		Flat Field Table Checksums
	0000000000001001		Command Status Return
	0000000000001010		Single Event Upset Information
	0000000000001011		Paged Region Memory Dump
	0000000000001100		Fixed Region Memory Dump
	0000000000001101		Software Performance Counters
	0000000000001110		Housekeeping Test Pattern
	0000000000001111		Spacecraft Time and Attitude Parameters
	xxxxxxxxxxxx0000		Spare Blocks
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3−31			Variable packet data dependant on the type identifier

3.1.1 Camera Observation Parameters

During normal observation modes, a number of 4x4 bins from each image are sampled, and stored in these housekeeping packets − one for each camera. This allows a quick−look facility to monitor the CCD performance without needing to fully decode the science data stream. Also in these packets are the 'critical observation parameters', which are used when decoding each image.

Word	Mask	Mnemonic	Parameter
0	000000000000xxxx	SOH_TYPE	Camera Observation Parameters
	0000000000000100		Camera 1
	0000000000000101		Camera 2
	0000000000000110		Camera 3
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxxxxxxxxxx	OBS_FRAME	Frame Number
4	xxxxxxxx−−−−−−−−	OBS_INTV	Frame Period (seconds)
4	−−−−−−−−xxxxxxxx	OBS_EXP	Frame Exposure Time (seconds)

5	--------xxxxxxxx	OBS_MODE	Observation mode
	--------00000000		Not currently observing
	--------00000001		Normal Observing Mode
	--------00000010		High Resolution Mode
	--------00000100		Engineering Mode
5	xxxxxxxx--------		Spare
6	--------0000xxxx	RICE_NB	Rice Compression Number of Noise Bits (0−15)
6	-------x--------	RICE_EN	Rice Compression Enabled
6	------x---------	RICE_DT	Rice Compression Delta Coding Enabled
6	-----x----------	FLAT_EN	Flat Field Correction Enabled
6	xxxxx-----------		Spare
7−15	xxxxxxxxxxxxxxxx		<TBD> Other Critical Observation Data
16	xxxxxxxxxxxxxxxx	CCD_BIN0	CCD Bin 0
17	xxxxxxxxxxxxxxxx	CCD_BIN1	CCD Bin 1
18	xxxxxxxxxxxxxxxx	CCD_BIN2	CCD Bin 2
19	xxxxxxxxxxxxxxxx	CCD_BIN3	CCD Bin 3
20	xxxxxxxxxxxxxxxx	CCD_BIN4	CCD Bin 4
21	xxxxxxxxxxxxxxxx	CCD_BIN5	CCD Bin 5
22	xxxxxxxxxxxxxxxx	CCD_BIN6	CCD Bin 6
23	xxxxxxxxxxxxxxxx	CCD_BIN7	CCD Bin 7
24	xxxxxxxxxxxxxxxx	CCD_BIN8	CCD Bin 8
25	xxxxxxxxxxxxxxxx	CCD_BIN9	CCD Bin 9
26	xxxxxxxxxxxxxxxx	CCD_BIN10	CCD Bin 10
27	xxxxxxxxxxxxxxxx	CCD_BIN11	CCD Bin 11
28	xxxxxxxxxxxxxxxx	CCD_BIN12	CCD Bin 12
29	xxxxxxxxxxxxxxxx	CCD_BIN13	CCD Bin 13
30	xxxxxxxxxxxxxxxx	CCD_BIN14	CCD Bin 14
31	xxxxxxxxxxxxxxxx	CCD_BIN15	CCD Bin 15

3.1.2 Analogue and Digital Monitors

- The analogue monitoring on SMEI is done using an 8−bit ADC
- See Appendix D for analogue monitor calibration factors.

Word	Mask	Mnemonic	Parameter
0	0000000000000111	SOH_TYPE	Analogue and Digital Monitors
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxx--------	AM_SUPP_I	AMon 0 : SMEI Current Monitor
3	--------xxxxxxxx	AM_DHU_5V	AMon 1 : Main DHU 5V supply
4	xxxxxxxx--------	AM_PROC_T	AMon 2 : Processor temperature monitor
4	--------xxxxxxxx	AM_PSU_T	AMon 3 : Power supply temperature monitor
5	xxxxxxxx--------	AM_SPARE	AMon 4 : Spare Analogue Monitor
5	--------xxxxxxxx	AM_PROC_I	AMon 5 : Processor current monitor
6	xxxxxxxx--------	AM_C1RAD_T	AMon 6 : Camera 1 Radiator Temperature
6	--------xxxxxxxx	AM_C1CCD_T	AMon 7 : Camera 1 CCD Temperature
7	xxxxxxxx--------	AM_C1EL_T	AMon 8 : Camera 1 Electronics Temperature
7	--------xxxxxxxx	AM_C1MIR_T	AMon 9 : Camera 1 Mirror Temperature
8	xxxxxxxx--------	AM_C1BAF_T	AMon 10 : Camera 1 Baffle Temperature
8	--------xxxxxxxx	AM_C1SPR_T	AMon 11 : Camera 1 Spare Temperature Monitor
9	xxxxxxxx--------	AM_C2RAD_T	AMon 12 : Camera 2 Radiator Temperature
9	--------xxxxxxxx	AM_C2CCD_T	AMon 13 : Camera 2 CCD Temperature
10	xxxxxxxx--------	AM_C2EL_T	AMon 14 : Camera 2 Electronics Temperature
10	--------xxxxxxxx	AM_C2MIR_T	AMon 15 : Camera 2 Mirror Temperature
11	xxxxxxxx--------	AM_C2BAF_T	AMon 16 : Camera 2 Baffle Temperature
11	--------xxxxxxxx	AM_C2SPR_T	AMon 17 : Camera 2 Spare Temperature Monitor
12	xxxxxxxx--------	AM_C3RAD_T	AMon 18 : Camera 3 Radiator Temperature
12	--------xxxxxxxx	AM_C3CCD_T	AMon 19 : Camera 3 CCD Temperature
13	xxxxxxxx--------	AM_C3EL_T	AMon 20 : Camera 3 Electronics Temperature
13	--------xxxxxxxx	AM_C3MIR_T	AMon 21 : Camera 3 Mirror Temperature
14	xxxxxxxx--------	AM_C3BAF_T	AMon 22 : Camera 3 Baffle Temperature
14	--------xxxxxxxx	AM_C3SPR_T	AMon 23 : Camera 3 Spare Temperature Monitor

15	xxxxxxxxxxxxxxxx		Digital Monitors 0
	−−−−−−−−xxxxxxxx	ADC_RB	ADC Conversion Value
	−−−xxxxx−−−−−−−−	ADC_MUX	ADC Multiplexer
	−−x−−−−−−−−−−−−−	ADC_WR	ADC WR Line Status
	−x−−−−−−−−−−−−−−	E2PROM_WEN	E^2Prom Write Enable
	x−−−−−−−−−−−−−−−	WDOG_EN	Watchdog Enable
16	xxxxxxxxxxxxxxxx		Digital Monitors 1
	−−−−−−−−−−−−−−−x	C1_SHT_OPN	Camera 1 Shutter (1 = Open)
	−−−−−−−−−−−−−−x−	C1_DOR_CLS	Camera 1 Door (1 = Closed)
	−−−−−−−−−−−−−x−−	C2_SHT_OPN	Camera 2 Shutter (1 = Open)
	−−−−−−−−−−−−x−−−	C2_DOR_CLS	Camera 2 Door (1 = Closed)
	−−−−−−−−−−−x−−−−	C3_SHT_OPN	Camera 3 Shutter (1 = Open)
	−−−−−−−−−−x−−−−−	C3_DOR_CLS	Camera 3 Door (1 = Closed)
	−−−−−−−−−x−−−−−−	C1_BOS_SUN	Camera 1 Bright Object Sensor (1 = Sun in view)
	−−−−−−−−x−−−−−−−	C2_BOS_SUN	Camera 2 Bright Object Sensor (1 = Sun in view)
	−−−−−−−x−−−−−−−−	C3_BOS_SUN	Camera 3 Bright Object Sensor (1 = Sun in view)
	−−−−−−x−−−−−−−−−	C3_1HZ	1Hz Monitor
	−−−xxx−−−−−−−−−−	CODE_VER	E2Prom Software Boot Code Identifier
	−−x−−−−−−−−−−−−−		Repeat of BANK1_SEL, bit 5
	−x−−−−−−−−−−−−−−	BOOT_RES	Boot Res
	x−−−−−−−−−−−−−−−	IRQ4	Interrupt 4
17	xxxxxxxxxxxxxxxx		Digital Monitors 2
	−−−−−−−−−xxxxxxx	BANK0_SEL	16K Memory Bank 0 Selector
	−−−−−−−−x−−−−−−−		Spare
	−xxxxxxx−−−−−−−−	BANK1_SEL	16K Memory Bank 1 Selector
	x−−−−−−−−−−−−−−−	AB_IDENT	A/B Processor Identifier
18	xxxxxxxxxxxxxxxx		Digital Monitors 3
	−xxxxxxxxxxxxxxx		Spare
	x−−−−−−−−−−−−−−−	AC_PARITY	FPGA 16−bit Parity Generator
19	xxxxxxxxxxxxxxxx		Digital Monitors 4
	−−−−−−−−−−−−−−−x	C1_PHASE_0	Camera 1 Shutter Phase 0 (Closed, No FF)
	−−−−−−−−−−−−−−x−	C1_PHASE_1	Camera 1 Shutter Phase 1 (Open, Hall B)
	−−−−−−−−−−−−−x−−	C1_PHASE_2	Camera 1 Shutter Phase 2 (Closed, FF)
	−−−−−−−−−−−−x−−−	C1_PHASE_3	Camera 1 Shutter Phase 3 (Open, Hall A)
	−−−−−−−−−−−x−−−−	C1_ON	Camera 1 On (FET control)
	−−−−−−−−−−x−−−−−	C1_SPARE1	Camera 1 Spare
	−−−−−−−−−x−−−−−−	C1_DEI_ON	Camera 1 De−ice Heater On
	−−−−−−−−x−−−−−−−	C1_HOP_ON	Camera 1 HOP On
	−−−−−−−x−−−−−−−−	C1_RLY_ON	Camera 1 Power Relay Coil Monitor
	−−−−−−x−−−−−−−−−	C1_SDA_IN	Camera 1 SDA In Monitor
	−−−−−x−−−−−−−−−−	C1_INT	Camera 1 Interrupt Status Monitor
	−−−−x−−−−−−−−−−−	C1_OVF	Camera 1 Hardware FIFO Overflow
	−−−x−−−−−−−−−−−−	C1_HOP_EN	Camera 1 HOP Protection Status
	−−x−−−−−−−−−−−−−	C1_SPARE2	Camera 1 Spare
	−x−−−−−−−−−−−−−−	C1_SPARE3	Camera 1 Spare
	x−−−−−−−−−−−−−−−	C1_LED_ON	Camera 1 LED On
20	xxxxxxxxxxxxxxxx		Digital Monitors 5
	−−−−−−−−−−−−−−−x	C2_PHASE_0	Camera 2 Shutter Phase 0 (Closed, No FF)
	−−−−−−−−−−−−−−x−	C2_PHASE_1	Camera 2 Shutter Phase 1 (Open, Hall B)
	−−−−−−−−−−−−−x−−	C2_PHASE_2	Camera 2 Shutter Phase 2 (Closed, FF)
	−−−−−−−−−−−−x−−−	C2_PHASE_3	Camera 2 Shutter Phase 3 (Open, Hall A)
	−−−−−−−−−−−x−−−−	C2_ON	Camera 2 On (FET control)
	−−−−−−−−−−x−−−−−	C2_SPARE1	Camera 2 Spare
	−−−−−−−−−x−−−−−−	C2_DEI_ON	Camera 2 De−ice Heater On
	−−−−−−−−x−−−−−−−	C2_HOP_ON	Camera 2 HOP On
	−−−−−−−x−−−−−−−−	C2_RLY_ON	Camera 2 Power Relay Coil Monitor
	−−−−−−x−−−−−−−−−	C2_SDA_IN	Camera 2 SDA In Monitor
	−−−−−x−−−−−−−−−−	C2_INT	Camera 2 Interrupt Status Monitor
	−−−−x−−−−−−−−−−−	C2_OVF	Camera 2 Hardware FIFO Overflow
	−−−x−−−−−−−−−−−−	C2_HOP_EN	Camera 2 HOP Protection Monitor
	−−x−−−−−−−−−−−−−	C2_SPARE2	Camera 2 Spare
	−x−−−−−−−−−−−−−−	C2_SPARE3	Camera 2 Spare
	x−−−−−−−−−−−−−−−	C2_LED_EN	Camera 2 LED On

21	xxxxxxxxxxxxxxxx		Digital Monitors 6
	---------------x	C3_PHASE_0	Camera 3 Shutter Phase 0 (Closed, No FF)
	--------------x-	C3_PHASE_1	Camera 3 Shutter Phase 1 (Open, Hall B)
	-------------x--	C3_PHASE_2	Camera 3 Shutter Phase 2 (Closed, FF)
	------------x---	C3_PHASE_3	Camera 3 Shutter Phase 3 (Open, Hall A)
	-----------x----	C3_ON	Camera 3 On
	----------x-----	C3_SPARE1	Camera 3 Spare
	---------x------	C3_DEI_ON	Camera 3 De−ice Heater On
	--------x-------	C3_HOP_ON	Camera 3 HOP On
	-------x--------	C3_RLY_ON	Camera 3 Power Relay Coil Monitor
	------x---------	C3_SDA_IN	Camera 3 SDA In Monitor
	-----x----------	C3_INT	Camera 3 Interrupt Status Monitor
	----x-----------	C3_OVF	Camera 3 Hardware FIFO Overflow
	---x------------	C3_HOP_EN	Camera 3 HOP Protection Monitor
	--x-------------	C3_SPARE2	Camera 3 Spare
	-x--------------	C3_SPARE3	Camera 3 Spare
	x---------------	C3_LED_ON	Camera 3 LED On
22	xxxxxxxxxxxxxxxx		Digital Monitors 7
	-----------xxxxx	WDOG_MSB	Watchdog Timer Eradicable
	--------000-----		Spare
	-------x--------	OBS_STAT	Current observing status
	------x---------	AB_1553	1553 A / B side selection status
	-----x----------	HOT_START	Hot start status
	----x-----------	WARM_START	Warm start status
	---x------------	1553_SSFLG	1553 SSFlag
	--x-------------	1553_INCMD	1553 In Command
	-x--------------	1553_MEMEN	1553 Memory Enable
	x---------------	1553_RDYD	1553 ReadyD
23	----------xxxxxx		Shutter motor phase readback
	--------------xx	C1_LSTPH	Last phase energized for Camera 1 (0−3)
	------------xx--	C2_LSTPH	Last phase energized for Camera 2 (0−3)
	----------xx----	C3_LSTPH	Last phase energized for Camera 3 (0−3)
24−31	xxxxxxxxxxxxxxxx		Spare (0)

3.1.3 Flat Field Table Checksums

The flat field tables used during camera data processing are periodically checked for single event errors using a simple xor−based checksum. The 16−bit checksums are returned in this packet.

Word	Mask	Mnemonic	Parameter
0	0000000000001000	SOH_TYPE	Flat Field Table Checksums
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxxxxxxxxxx	CSUM_BLK0	Block checksum 0
4	xxxxxxxxxxxxxxxx	CSUM_BLK1	Block checksum 1
5	xxxxxxxxxxxxxxxx	CSUM_BLK2	Block checksum 2
.			
.			
.			
29	xxxxxxxxxxxxxxxx	CSUM_BLK26	Block checksum 26
30	xxxxxxxxxxxxxxxx	CSUM_BLK27	Block checksum 27
31	xxxxxxxxxxxxxxxx	CSUM_BLK28	Block checksum 28

3.1.4 Command / Instrument Status Return

All 1553 messages received by SMEI on subaddresses 4 and 5 have status information returned in this packet. The table is filled in a cyclic fashion, and a counter identifies the last filled entry. The SMEI global configuration word and mode are also returned in this packet.

Word	Mask	Mnemonic	Parameter
0	0000000000001001	SOH_TYPE	Command Status Return
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet

3	xxxxxxxx−−−−−−−−	CMD_ACP	Total number of valid commands accepted
3	−−−−−−−−xxxxxxxx	CMD_REJ	Total number of invalid commands rejected
4	xxxxxxxxxxxxxxxx	SOH_TOT	Total number of SOH requests serviced
5	xxxxxxxx−−−−−−−−	SCI_TOT	Total number of science data requests serviced
5	−−−−−−−−xxxxxxxx	ATT_TOT	Total number of attitude data packets received
6	xxxxxxxxxxxxxxxx	SMEI_CONF	Global Instrument Configuration Word
	−−−−−−−−−−−−−−−0		Disable 1553 test mode
	−−−−−−−−−−−−−−−1		Enable 1553 test mode
	−−−−−−−−−−−−−−0−		Use ramp pattern for all 1553 test mode output
	−−−−−−−−−−−−−−1−		Use fixed pattern for all 1553 test mode output
	−−−−−−−−−−−−−0−−		Disable science data test mode
	−−−−−−−−−−−−−1−−		Enable science data test mode
	−−−−−−−−−−−−0−−−		Use ramp pattern for science data test mode pixels
	−−−−−−−−−−−−1−−−		Use fixed value for science data test mode pixels
	xxxxxxxxxxxx−−−−		Spare
7	xxxx−−−−−−−−−−−−	SMEI_MTI	SMEI Internal Mode Control and Option Flags
	−−−−xxxx−−−−−−−−	SMEI_MODE	Current Instrument Operating Mode
	−−−−0000−−−−−−−−		Boot mode
	−−−−0001−−−−−−−−		Configuration mode
	−−−−0010−−−−−−−−		Patch mode
	−−−−0011−−−−−−−−		Safe mode
	−−−−0100−−−−−−−−		Observation mode
7	−−−−−−−−00000xxx	CMD_LAST	Number of last entry filled (0−7)
8	xxxxxxxxxxxxxxxx	CMD0_ID	#0 Command Identifier
9	xxxxxxxxxxxxxxxx	CMD0_CS	#0 Command Checksum (CRC)
10	xxxxxxxxxxxxxxxx	CMD0_ST	#0 Command Status
	−−−−−−−−−−−−−−−1		Command was received and executed correctly
	−−−−−−−−−−−−−−1−		Command had a CRC error and was not executed
	−−−−−−−−−−−−−1−−		Command was illegally formed (not recognised)
	−−−−−−−−−−−−1−−−		The instrument mode did not permit the command
	−−−−−−−−−−−1−−−−		The (secure) command had not been enabled
	−−−−−−−−−−1−−−−−		Message received on invalid sub−address
	−−−−−−−−−1−−−−−−		Message errors flagged by interface hybrid
	−−−−−−−−1−−−−−−−		Message sequence number was not correct
	−−−−−−−x−−−−−−−−	CMD0_BUS	1553 Bus Message was received on (0=A, 1=B)
	xxxxxxx−−−−−−−−−		Message specific error codes
.			
.			
29	xxxxxxxxxxxxxxxx	CMD7_ID	#7 Command Identifier
30	xxxxxxxxxxxxxxxx	CMD7_CS	#7 Command Checksum
31	xxxxxxxxxxxxxxxx	CMD7_ST	#7 Command Status
	−−−−−−−−−−−−−−−1		Command was received and executed correctly
	−−−−−−−−−−−−−−1−		Command had a CRC error and was not executed
	−−−−−−−−−−−−−1−−		Command was illegally formed (not recognised)
	−−−−−−−−−−−−1−−−		The instrument mode did not permit the command
	−−−−−−−−−−−1−−−−		The (secure) command had not been enabled
	−−−−−−−−−−1−−−−−		Message received on invalid sub−address
	−−−−−−−−−1−−−−−−		Message errors flagged by 1553 interface chip
	−−−−−−−−1−−−−−−−		Message sequence number was not correct
	−−−−−−−x−−−−−−−−	CMD7_BUS	1553 Bus Message was received on (0=A, 1=B)
	xxxxxxx−−−−−−−−−		Message specific error codes

The command identifier uniquely identifies the command type, and the checksum is used to distinguish individual commands.

The status word provides details of the actions SMEI took in response to the command.

The last eight commands which were received are stored in CMD***x***_ID/CS/ST. This array is treated as a circular buffer, and CMD_LAST indicates the most recently filled entry, ie ***x***.

3.1.5 Single Event Upset Information

SMEI has regions of memory allocated for three copies of critical instrument parameters and switches. These areas are routinely monitored for single event upsets by mutual−comparison, and information about any anomalies found are reported in this housekeeping block. SMEI uses per−bit correction, and therefore a value can never be unable to be corrected, as at least two of the three bits will always match. The array is treated as a circular buffer, and TRM_LAST indicates the most recently filled entry in the array.

Word	Mask	Mnemonic	Parameter
0	0000000000001010	SOH_TYPE	Single event upset information
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxxxxxxxxxx	TRM_CERR	Total number of correctable upsets recorded
4	xxxxxxxx		Spare
4	--------0000xxxx	TRM_LAST	Number of last entry filled (0−8)
5	xxxxxxxxxxxxxxxx	TRM0_ADDR	#0 Address
6	xxxxxxxxxxxxxxxx	TRM0_TS	#0 Time−stamp
7	xxxxxxxxxxxxxxxx	TRM0_RV	#0 Replacement Value
8	xxxxxxxxxxxxxxxx	TRM1_ADDR	#1 Address
9	xxxxxxxxxxxxxxxx	TRM1_TS	#1 Time−stamp
10	xxxxxxxxxxxxxxxx	TRM1_RV	#1 Replacement Value
11	xxxxxxxxxxxxxxxx	TRM2_ADDR	#2 Address
12	xxxxxxxxxxxxxxxx	TRM2_TS	#2 Time−stamp
13	xxxxxxxxxxxxxxxx	TRM2_RV	#2 Replacement Value
14	xxxxxxxxxxxxxxxx	TRM3_ADDR	#3 Address
15	xxxxxxxxxxxxxxxx	TRM3_TS	#3 Time−stamp
16	xxxxxxxxxxxxxxxx	TRM3_RV	#3 Replacement Value
17	xxxxxxxxxxxxxxxx	TRM4_ADDR	#4 Address
18	xxxxxxxxxxxxxxxx	TRM4_TS	#4 Time−stamp
19	xxxxxxxxxxxxxxxx	TRM4_RV	#4 Replacement Value
20	xxxxxxxxxxxxxxxx	TRM5_ADDR	#5 Address
21	xxxxxxxxxxxxxxxx	TRM5_TS	#5 Time−stamp
22	xxxxxxxxxxxxxxxx	TRM5_RV	#5 Replacement Value
23	xxxxxxxxxxxxxxxx	TRM6_ADDR	#6 Address
24	xxxxxxxxxxxxxxxx	TRM6_TS	#6 Time−stamp
25	xxxxxxxxxxxxxxxx	TRM6_RV	#6 Replacement Value
26	xxxxxxxxxxxxxxxx	TRM7_ADDR	#7 Address
27	xxxxxxxxxxxxxxxx	TRM7_TS	#7 Time−stamp
28	xxxxxxxxxxxxxxxx	TRM7_RV	#7 Replacement Value
29	xxxxxxxxxxxxxxxx	TRM8_ADDR	#8 Address
30	xxxxxxxxxxxxxxxx	TRM8_TS	#8 Time−stamp
31	xxxxxxxxxxxxxxxx	TRM8_RV	#8 Replacement Value

3.1.6 Paged Region Memory Dump

This is a simple sliding dump of the contents of the E^2Prom and bulk storage SRAM. This allows a slow−scan picture of the current state of the entire SMEI memory to be built. Single bit errors can be located and corrected via patching as required.

Word	Mask	Mnemonic	Parameter
0	0000000000001011	SOH_TYPE	Paged Region Memory Dump
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxx--------		Spare
3	--------0xxxxxxx	MEM_PAGE	Page selector
4	00xxxxxxxxxxxxxx	MEM_OFFSET	Page start offset (0 − 3FE5h)
5−31	xxxxxxxxxxxxxxxx	MEM_DATA	Memory dump

3.1.7 Fixed Region Memory Dump

This is a simple sliding dump of the contents of the processor RAM. This allows a slow−scan picture of the current state of the memory to be built. Single bit errors can be located and corrected via patching as required.

Word	Mask	Mnemonic	Parameter
0	0000000000001100	SOH_TYPE	Fixed Memory Dump
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TYPE	Time−stamp of the last update of this data packet
3	0xxxxxxxxxxxxxxx	FIX_OFFSET	Page start offset (0 − 7FE4h)
4−31	xxxxxxxxxxxxxxxx	FIX_DATA	Memory dump

3.1.8 Software Performance Counters

The onboard software has a number of performance counters built in, which allow bottlenecks to be detected. These include high and low watermarks for buffer usage, pixel processing counts, interrupt totals, uptime counter, mode, SEU counts, and so forth.

<**TBC/TBD** until software performance monitoring is finalised>

Word	Mask	Mnemonic	Parameter
0	0000000000001101	SOH_TYPE	Software Performance Counters
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3	xxxxxxxxxxxxxxxx	INT1_MSW	Count of Camera 1 FIFO Interrupts
4	xxxxxxxxxxxxxxxx	INT1_LSW	Count of Camera 1 FIFO Interrupts
5	xxxxxxxxxxxxxxxx	INT2_MSW	Count of Camera 2 FIFO Interrupts
6	xxxxxxxxxxxxxxxx	INT2_LSW	Count of Camera 2 FIFO Interrupts
7	xxxxxxxxxxxxxxxx	INT3_MSW	Count of Camera 3 FIFO Interrupts
8	xxxxxxxxxxxxxxxx	INT3_LSW	Count of Camera 3 FIFO Interrupts
9	xxxxxxxxxxxxxxxx	INT4_MSW	Count of 1Hz Interrupts
10	xxxxxxxxxxxxxxxx	INT4_LSW	Count of 1Hz Interrupts
11	xxxxxxxxxxxxxxxx	C1RB_HWM	Camera 1 High Watermark for IRQ ring buffer
12	xxxxxxxxxxxxxxxx	C2RB_HWM	Camera 2 High Watermark for IRQ ring buffer
13	xxxxxxxxxxxxxxxx	C3RB_HWM	Camera 3 High Watermark for IRQ ring buffer
14	xxxxxxxxxxxxxxxx	C1RB_OVF	Camera 1 Overflow counter for IRQ ring buffer
15	xxxxxxxxxxxxxxxx	C2RB_OVF	Camera 2 Overflow counter for IRQ ring buffer
16	xxxxxxxxxxxxxxxx	C3RB_OVF	Camera 3 Overflow counter for IRQ ring buffer
17−31	xxxxxxxxxxxxxxxx		Spare

3.1.9 Housekeeping Test Pattern

To enable verification of the X− and S−band channels, a simple test pattern − incrementing ramp −is available.

Word	Mask	Mnemonic	Parameter
0	0000000000001110	SOH_TYPE	Housekeeping Test Pattern
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3−31	xxxxxxxxxxxxxxxx	HTP_DATA	Each word is the previous HTP_DATA word + 0x0001

3.1.10 Spacecraft Time and Attitude Parameters

This packet provides time and attitude data for ground processing. Data received by SMEI from the spacecraft containing spacecraft time and attitude data is made available here. Note that words 2 and 3 from the data received by SMEI from the spacecraft are reversed in this housekeeping block.

Word	Mask	Mnemonic	Parameter
0	0000000000001111	SOH_TYPE	Time and Attitude Parameters
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	ICD Figure C6−3, **word 3** (SCT Seconds LSW)
3	xxxxxxxxxxxxxxxx	SCT_MSW	ICD Figure C6−3, **word 2** (SCT Seconds MSW)
4	xxxxxxxxxxxxxxxx	SCT_SSEC	ICD Figure C6−3, word 4
5	xxxxxxxxxxxxxxxx	FQ_Q1_MSW	ICD Figure C6−3, word 5
6	xxxxxxxxxxxxxxxx	FQ_Q1_LSW	ICD Figure C6−3, word 6
7	xxxxxxxxxxxxxxxx	FQ_Q2_MSW	ICD Figure C6−3, word 7
8	xxxxxxxxxxxxxxxx	FQ_Q2_LSW	ICD Figure C6−3, word 8
9	xxxxxxxxxxxxxxxx	FQ_Q3_MSW	ICD Figure C6−3, word 9
10	xxxxxxxxxxxxxxxx	FQ_Q3_LSW	ICD Figure C6−3, word 10
11	xxxxxxxxxxxxxxxx	FQ_Q4_MSW	ICD Figure C6−3, word 11
12	xxxxxxxxxxxxxxxx	FQ_Q4_LSW	ICD Figure C6−3, word 12
13	xxxxxxxxxxxxxxxx	IBE_X_MSW	ICD Figure C6−3, word 13
14	xxxxxxxxxxxxxxxx	IBE_X_LSW	ICD Figure C6−3, word 14
15	xxxxxxxxxxxxxxxx	IBE_Y_MSW	ICD Figure C6−3, word 15
16	xxxxxxxxxxxxxxxx	IBE_Y_LSW	ICD Figure C6−3, word 16
17	xxxxxxxxxxxxxxxx	IBE_Z_MSW	ICD Figure C6−3, word 17
18	xxxxxxxxxxxxxxxx	IBE_Z_LSW	ICD Figure C6−3, word 18
19	xxxxxxxxxxxxxxxx	TWM_X_MSW	ICD Figure C6−3, word 19
20	xxxxxxxxxxxxxxxx	TWM_X_LSW	ICD Figure C6−3, word 20
21	xxxxxxxxxxxxxxxx	TWM_Y_MSW	ICD Figure C6−3, word 21
22	xxxxxxxxxxxxxxxx	TWM_Y_LSW	ICD Figure C6−3, word 22
23	xxxxxxxxxxxxxxxx	TWM_Z_MSW	ICD Figure C6−3, word 23
24	xxxxxxxxxxxxxxxx	TWM_Z_LSW	ICD Figure C6−3, word 24
25	xxxxxxxxxxxxxxxx	SL_X_MSW	ICD Figure C6−3, word 25
26	xxxxxxxxxxxxxxxx	SL_X_LSW	ICD Figure C6−3, word 26
27	xxxxxxxxxxxxxxxx	SL_Y_MSW	ICD Figure C6−3, word 27
28	xxxxxxxxxxxxxxxx	SL_Y_LSW	ICD Figure C6−3, word 28
29	xxxxxxxxxxxxxxxx	SL_Z_MSW	ICD Figure C6−3, word 29
30	xxxxxxxxxxxxxxxx	SL_Z_LSW	ICD Figure C6−3, word 30
31	xxxxxxxxxxxxxxxx		ICD Figure C6−3, word 31 (spare)

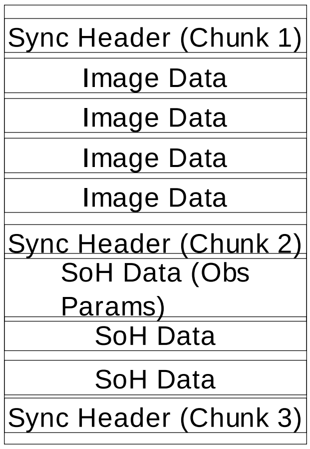
3.1.11 Reserved Blocks

These blocks are reserved for further SoH parameters, should they be needed. They have the same basic header structure as other SoH packets.

Word	Mask	Mnemonic	Parameter
0	xxxxxxxxxxxxxxxx	SOH_TYPE	Reserved Blocks
	0000000000000001		Spare
	0000000000000010		Spare
	0000000000000011		Spare
	xxxxxxxxxxxx0000		Spare
1	xxxxxxxxxxxxxxxx	SOH_CRC	Cyclic Redundancy Check
2	xxxxxxxxxxxxxxxx	SOH_TIME	Time−stamp of the last update of this data packet
3−31	xxxxxxxxxxxxxxxx		Spare

3.2 Science Data Stream

The science data stream is created by multiplexing image data and state of health data. A small header is used to delineate the different data types, and to allow the data processing software to resynchronise after a loss of telemetry. We define a synchronisation header, plus all the data until the next synchronisation header as a 'chunk'. Each chunk has a maximum size of 8192 words.



When embedding state of health information into the science data stream, the packet formats are identical to those used in the state of health data stream. These formats are defined in sections 3.1.1 to 3.1.11. A state of health chunk may hold one or more state of health packets.

When embedding camera image data into the science data stream, a complete frame from one camera is transmitted, in one or more back−to−back chunks containing image data packets. Immediately following the image data chunk(s), a chunk containing the camera observation parameter packet (section 3.1.1), and potentially other state of health packets is transmitted.

The figure to the right shows a sample data stream. In this example, a complete image (fitting completely into a single chunk) is followed by a set of housekeeping data in chunk 2. The first state of health packet in chunk 2 is the camera observation packet for the image, and in this example, is followed by 2 other state of health packets. The next chunk (3) can contain either image data, or more state of health data.

3.2.1 Science Data Stream Synchronisation Header

This is the simple header used to separate the science data stream into image and state of health data. A synchronisation word provides a mechanism to locate the header, and the size field allows rapid location of the next header in the data stream (and also verification that the synchronisation word located was not a false−positive).

When the header is followed by camera image data for a new image, the SH_TYPE field contains 0, and the SH_CAM field is used to to identify which camera image data is from. If there are too many image data packets to fit into a single 8192 word chunk, a new synchronisation header is inserted into the stream, with the SH_CAM field holding the image data continuation marker (00), and with SH_TYPE containing 0.

When the header is followed by state of health data, the SH_TYPE field is 1, and the SH_CAM field is 00.

Word	Mask	Mnemonic	Parameter
0	xxxxxxxxxxxxxxxx	SH_SYNC	Synchronisation Pattern
1	−−−xxxxxxxxxxxxx	SH_SIZE	Number of words until the next header
1	−xx−−−−−−−−−−−−−	SH_CAM	Camera Identifier or Continuation indicator
	−00−−−−−−−−−−−−−		Image Data Continuation Marker / State of Health
	−01−−−−−−−−−−−−−		Camera 1
	−10−−−−−−−−−−−−−		Camera 2
	−11−−−−−−−−−−−−−		Camera 3
1	x−−−−−−−−−−−−−−−	SH_TYPE	Image or Housekeeping indicator
	0−−−−−−−−−−−−−−−		Image Data
	1−−−−−−−−−−−−−−−		State of Health Data

3.2.2 Image Data Packet Format

Each frame of camera image data is buffered in memory until the entire frame has been processed. The image data is formatted into 264 word packets − 8 words of error correction code, followed by 256 words of (optionally rice compressed) image data.

Word	Mask	Mnemonic	Parameter
0	xxxxxxxxxxxxxxxx	IDP_ECC0	Error correction data
1	xxxxxxxxxxxxxxxx	IDP_ECC1	Error correction data
2	xxxxxxxxxxxxxxxx	IDP_ECC2	Error correction data
3	xxxxxxxxxxxxxxxx	IDP_ECC3	Error correction data
4	xxxxxxxxxxxxxxxx	IDP_ECC4	Error correction data
5	xxxxxxxxxxxxxxxx	IDP_ECC5	Error correction data
6	xxxxxxxxxxxxxxxx	IDP_ECC6	Error correction data
7	xxxxxxxxxxxxxxxx	IDP_ECC7	Error correction data
8−263	xxxxxxxxxxxxxxxx	IDP_DATA	Image data.

It is usual for the image data not to fit exactly into a whole number of 256 word packets. When the final packet of an image is filled with data, the remaining words of the packet are padded out with zeros when calculating the error correction codes.

This zero padding is *not transmitted* in the science data stream. The final image data packet for a frame of camera data is truncated. The length of this packet is derived by examining the SH_SIZE field of the preceding synchronisation header. For image data, the field contains (264 * full packets) + (size of last packet). There are always 8 words of error correction code.

The error correction code used here is a rectangular coding scheme. The diagram below shows the correspondence between the compressed image data, and the error correction words. IDP_ECC0−3 are calculated by a simple xor operation down each column. IDP_ECC4−7 are calculated from the parity bit for each row of the table. Further details are in Appendix A.

Word 0	Word 1	Word 2	Word 3	IDP_ECC4
. . .	. . .	. . .	. . .	
Word 60	Word 61	Word 62	Word 63	
Word 64	Word 65	Word 66	Word 67	IDP_ECC5
. . .	. . .	. . .	. . .	
Word 124	Word 125	Word 126	Word 127	
Word 128	Word 129	Word 130	Word 131	IDP_ECC6
. . .	. . .	. . .	. . .	
Word 188	Word 189	Word 190	Word 191	
Word 192	Word 193	Word 194	Word 195	IDP_ECC7
. . .	. . .	. . .	. . .	
Word 252	Word 253	Word 254	Word 255	
IDP_ECC0	IDP_ECC1	IDP_ECC2	IDP_ECC3	

A. Rectangular Error Correction Coding

Rectangular error correction codes work by arranging the data into an array of *m* x *n* bits. For each row and column of the array, a parity bit is generated, and these parity bits are included in the data transmitted. For each encoded packet, rectangular encoding can detect and correct single bit errors, and also detect all dual bit errors, and a number of other bit error patterns.

Decoding single bit errors requires each row and column be parity checked. The incorrect bit is located by cross−referencing the row and column for which the parity check failed.

A.1 SMEI Conventions

For image data, we chose to use an array of 64 x *N* bits for generating the error correction codes.

The final part of an image may not completely fill a 64 x *N* array, and so to avoid wasting bandwidth, the final array is transmitted as a 64 x *Q* array, with *Q* varying as needed. The error correction code generation is performed as if the 64 x *Q* block was 64 x *N*, with the unused words filled with zero.

The bit error rate of 10^{-6} specified in the ICD for data received at the ground station determines the rate at which we expect to receive image data which cannot be corrected. The following table shows a range of results for rectangular encoding.

Downlink BER	1E−06	P(error per block) = [P(downlink) * Total Block Bits] ^ Errors Per Block
Frame Size	63200 Bits	P(error per frame) = P(error per block) * Data Blocks Per Frame
	Assumes 2:1 Compression	Overhead = ECC bits / Data bits

	RECT 64x16bit	RECT 64x32bit	RECT 64x64bit	RECT 64x128bit
Block Data Bits	1024	2048	4096	8192
Block Data Bytes	128	256	512	1024
ECC Bits	80	96	128	192
Overhead	7.81%	4.69%	3.13%	2.34%

Errors/block	P(error)	1/P(error)	P(error)	1/P(error)	P(error)	1/P(error)	P(error)	1/P(error)
1	1.1E−03	906	2.1E−03	466	4.2E−03	237	8.4E−03	119
2	1.2E−06	820468	4.6E−06	217546	1.8E−05	56047	7.0E−05	14226
3	1.3E−09	743177889	9.9E−09	101467196	7.5E−08	13268697	5.9E−07	1696861

Errors/frame	P(error)	1/P(error)	P(error)	1/P(error)	P(error)	1/P(error)	P(error)	1/P(error)
1	6.8E−02	15	6.6E−02	15	6.5E−02	15	6.5E−02	15
2	7.5E−05	13294	1.4E−04	7050	2.8E−04	3632	5.4E−04	1844
3	8.3E−08	12041363	3.0E−07	3288051	1.2E−06	859946	4.5E−06	219948

We select *N* to be 64, as this gives a good trade−off between the overhead of the correction codes (3.2%), and the mean time between dual−bit errors.

If we assume a compression ratio of 2:1 for science image data, then in normal observation mode we expect to see one image in 3,600 with an image data packet containing a dual−bit error, or approximately one frame every 80 minutes.

For *N* = 64, there are 8 words of error correction data for every 256 words of compressed image data. In the image data packet (section 3.2.2), we define the error correction words as IDP_ECC0 to IDP_ECC7. They are calculated as follows:

IDP_ECC0	Xor data words 0, 4, 8, (w*4 + 0), 252.
IDP_ECC1	Xor data words 1, 5, 9, (w*4 + 1), 253.
IDP_ECC2	Xor data words 2, 6, 10, (w*4 + 2), 254.
IDP_ECC3	Xor data words 3, 7, 11, (w*4 + 3), 255.
IDP_ECC4	Bit *b* generated from parity bit for data words (b*4) to (b*4 + 3).
IDP_ECC5	Bit *b* generated from parity bit for data words (b*4 + 64) to (b*4 + 67).
IDP_ECC6	Bit *b* generated from parity bit for data words (b*4 + 128) to (b*4 + 131).
IDP_ECC7	Bit *b* generated from parity bit for data words (b*4 + 192) to (b*4 + 195).

B. Housekeeping Checksum Coding

The checksum coding used for SMEI housekeeping blocks is the same as used by WindSat. The code is generated using the standard 16−bit SDLC CRC algorithm, as defined in 'Numerical Recipes in C, Second Edition'. Included here is a copy of a routine written in C to calculate CRC−16.

```
#include <stdio.h>
#define bufsiz (16*1024)
static WORD crc_table[256];

void init_crc_table(void)
{
      int i, j;
      WORD k;

      for (i = 0; i < 256; i++)
      {
            k = 0xC0C1;
            for (j = 1; j < 256; j <<= 1)
            {
                  if (i & j)
                        crc_table[i] ^= k;
                  k = (k << 1) ^ 0x4003;
            }
      }
}

/* crc_calc() −− calculate cumulative crc−16 for buffer */
WORD crc_calc(WORD crc, char *buf, unsigned nbytes)
{
      unsigned char *p, *lim;

      p = (unsigned char *)buf;
      lim = p + nbytes;
      while (p < lim)
            crc = (crc >> 8 ) ^ crc_table[(crc & 0xFF) ^ *p++];

      return crc;
}

void do_file(char *fn)
{
      static char buf[bufsiz];
      FILE *f;
      int k;
      WORD crc;

      f = fopen(fn, "rb");
      if (f == NULL)
      {
            printf("%s: can't open file\n", fn);
            return;
      }
      crc = 0;
      while ((k = fread(buf, 1, bufsiz, f)) != 0)
            crc = crc_calc(crc, buf, k);
      fclose(f);
      printf("%−14s %04X\n", fn, crc);
}

int main(int argc, char **argv)
{
      int i;

      if (argc < 2)
      {
            fprintf(stderr, "Usage: crc filename [filename...]\n");
            return EXIT_FAILURE;
      }
      init_crc_table();
      for (i = 1; i < argc; i++)
            do_file(argv[i]);
      return EXIT_SUCCESS;
}
```

C. Rice Compression Coding

The compression scheme used for SMEI image data is the Rice Compression Scheme. This scheme extracts the noise bits from the data, and sends these bits 'as−is'. The remainder of the word is transmitted using difference−encoding, with special codes used for extreme jumps in value. Rice compression is a lossless algorithm.

The scheme used is documented in a paper by Michael W. Richmond and Nancy E. Ellman, titled 'Another Technique for Compressing Astronomical Imaging'. The paper and sample source code is available on the web.

- Original http://stupendous.isc.rit.edu/richmond/rice/
- Local copy http://www.sr.bham.ac.uk/~mpc/pulsar/smei/rice/

D. Analogue Monitor Calibration Tables

There are 24 analogue monitors on SMEI, as defined in the following table.

Amon	**Function**	**Sensor**	**Range**	
0	Instrument Current	Linear	0.000A	3.840A
1	Proc PCB 5V	Linear	3.559v	6.355v
2	Proc PCB Temperature	YSI 44004	70C	−30C
3	DHU Power Supply Temperature	YSI 44004	70C	−30C
4	Spare			
5	Proc PCB Current	Linear	0.000A	1.500A
6	Camera 1 Radiator Temperature	YSI 44003A	40C	−60C
7	Camera 1 CCD Temperature	YSI 44003A	40C	−60C
8	Camera 1 Electronics Temperature	YSI 44004	70C	−30C
9	Camera 1 Mirror Temperature	YSI 44004	70C	−30C
10	Camera 1 Baffle Temperature	YSI 44003A	40C	−60C
11	Camera 1 Cold Finger Temperature	YSI 44004	70C	−30C
12	Camera 2 Radiator Temperature	YSI 44003A	40C	−60C
13	Camera 2 CCD Temperature	YSI 44003A	40C	−60C
14	Camera 2 Electronics Temperature	YSI 44004	70C	−30C
15	Camera 2 Mirror Temperature	YSI 44004	70C	−30C
16	Camera 2 Baffle Temperature	YSI 44003A	40C	−60C
17	Camera 2 Cold Finger Temperature	YSI 44004	70C	−30C
18	Camera 3 Radiator Temperature	YSI 44003A	40C	−60C
19	Camera 3 CCD Temperature	YSI 44003A	40C	−60C
20	Camera 3 Electronics Temperature	YSI 44004	70C	−30C
21	Camera 3 Mirror Temperature	YSI 44004	70C	−30C
22	Camera 3 Baffle Temperature	YSI 44003A	70C	−30C
23	Camera 3 Cold Finger Temperature	YSI 44004	70C	−30C

The current and voltage monitors are linear inputs, and a simple offset plus scaling is all that is required to convert from ADC units into engineering units.

The situation for the non−linear thermistors is more complex. For a given thermistor resistance, the temperature can be determined using:

$$\frac{1}{T} = a + b(\ln R_t) + c(\ln R_t)^3 \qquad (1)$$

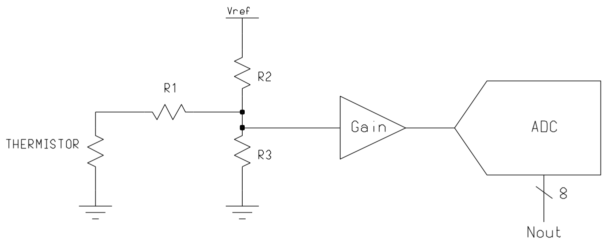
where: T = Temperature in Kelvin

a,b,c = coefficients derived from measurement of the thermistor type (see table below)

ln R_t = natural logarithm of the resistance in ohms

Thermistor	**a**	**b**	**c**
YSI44004 (Mix B)	0.0014733	0.0002372	1.07E−007
YSI44003A (Mix L)	0.0013130	0.0002906	1.02E−007

The thermistors are arranged in a simple potential divider arrangement, and the following equations define the relationship between the resistance of the thermistor (R_t), combined resistance of R_t, R_1 and R_3 (R_p), voltage presented to the ADC (v_o), and the final convertor output (n_{out}).



In general terms:

$$R_p = \frac{((R_t + R_1) x R_3)}{((R_t + R_1) + R_3)}$$

$$v_o = \left[\frac{(2.5 x R_p)}{(R_2 + R_p)} \right] x 3.49 \qquad (2)$$

$$n_{out} = \frac{(v_o - 1.694)}{(5.2 x 10^{-3})}$$

where: R_p =Resistance of the R_t, R_1, R_3 network in ohms

R_t = Thermistor resistance in ohms

v_o = Input voltage to the ADC in volts

n_{out} = Converted value (0−255)

Gain = 3.49

For YSI44004:

R1 = 953 Ω

R2 = 3240 Ω

R3 = 1820 Ω

For YSI44003A:

R1 = 2000 Ω

R2 = 6190 Ω

R3 = 3480 Ω

Once R_t has been calculated, the temperature can be determined using equation (1).