# SMEI Instrument Telemetry Format Specification

**M.P.Cooke**

and

**C.J.Eyles**

**University of Birmingham**

Issue 1

25 Jan 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 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.

## 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 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 GPS 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 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 GPS seconds data (UTC Seconds LSW) that the spacecraft supplies to SMEI periodically. This allows good knowledge of when the housekeeping was last updated.

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | xxxxxxxxxxxxxxxx | SOH_TYPE | Multiplex Packet Type Identifier |
| | 0000000000000001 | | Reserved Block |
| | 0000000000000010 | | Reserved Block |
| | 0000000000000100 | | Reserved Block |
| | 0000000000001000 | | Camera 1 Observation Parameters |
| | 0000000000010000 | | Camera 2 Observation Parameters |
| | 0000000000100000 | | Camera 3 Observation Parameters |
| | 0000000001000000 | | Analogue and Digital Monitors |
| | 0000000010000000 | | Flat Field Table Checksums |
| | 0000000100000000 | | Command Status Return |
| | 0000001000000000 | | Single Event Upset Information |
| | 0000010000000000 | | Paged Region Memory Dump |
| | 0000100000000000 | | Fixed Region Memory Dump |
| | 0001000000000000 | | Software Performance Counters |
| | 0010000000000000 | | Housekeeping Test Pattern |
| | 0100000000000000 | | GPS Time and Attitude Parameters |
| | 1000000000000000 | | Reserved Block |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp 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 pixels 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 | 0000000000000xxx | SOH_TYPE | Camera Observation Parameters |
| | 0000000000001000 | | Camera 1 |
| | 0000000000010000 | | Camera 2 |
| | 0000000000100000 | | Camera 3 |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3 | xxxxxxxxxxxxxxxx | PX_PRE0 | Pre−image Pixel 0 |
| 4 | xxxxxxxxxxxxxxxx | PX_PRE1 | Pre−image Pixel 1 |
| 5 | xxxxxxxxxxxxxxxx | PX_PRE2 | Pre−image Pixel 2 |

| | | | |
|---|---|---|---|
| 6 | xxxxxxxxxxxxxxxx | PX_PRE3 | Pre−image Pixel 3 |
| 7 | xxxxxxxxxxxxxxxx | PX_POST0 | Post−image Pixel 0 |
| 8 | xxxxxxxxxxxxxxxx | PX_POST1 | Post−image Pixel 1 |
| 9 | xxxxxxxxxxxxxxxx | PX_POST2 | Post−image Pixel 2 |
| 10 | xxxxxxxxxxxxxxxx | PX_POST3 | Post−image Pixel 3 |
| 11 | xxxxxxxxxxxxxxxx | OBS_FRAME | Frame Number |
| 12 | xxxxxxxx-------- | OBS_INTV | Frame Period (seconds) |
| 12 | --------xxxxxxxx | OBS_EXP | Frame Exposure Time (seconds) |
| 13 | --------xxxxxxxx | OBS_MODE | Observation mode |
| | --------00000000 | | Not currently observing |
| | --------00000001 | | Normal Observing Mode |
| | --------00000010 | | High Resolution Mode |
| | --------00000100 | | Engineering Mode |
| 13 | xxxxxxxx-------- | | Spare |
| 14 | --------0000xxxx | RICE_NB | Rice Compression Number of Noise Bits (0−15) |
| 14 | -------x-------- | RICE_EN | Rice Compression Enabled |
| 14 | ------x--------- | RICE_DT | Rice Compression Delta Coding Enabled |
| 14 | xxxxxx---------- | | Spare |
| 15−23 | xxxxxxxxxxxxxxxx | | <TBD> Other Critical Observation Data |
| 24 | xxxxxxxxxxxxxxxx | ROV0_ID | Roving bin 0 identifier |
| 25 | xxxxxxxxxxxxxxxx | ROV0_AMP | Roving bin 0 amplitude |
| 26 | xxxxxxxxxxxxxxxx | ROV1_ID | Roving bin 1 identifier |
| 27 | xxxxxxxxxxxxxxxx | ROV1_AMP | Roving bin 1 amplitude |
| 28 | xxxxxxxxxxxxxxxx | ROV2_ID | Roving bin 2 identifier |
| 29 | xxxxxxxxxxxxxxxx | ROV2_AMP | Roving bin 2 amplitude |
| 30 | xxxxxxxxxxxxxxxx | ROV3_ID | Roving bin 3 identifier |
| 31 | xxxxxxxxxxxxxxxx | ROV3_AMP | Roving bin 3 amplitude |

### 3.1.2 Analogue and Digital Monitors

- The analogue monitoring on SMEI is done using an 8−bit ADC
- Calibration details will not be available until the flight hardware is available. See Appendix D for analogue monitor conversion factors.

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | 0000000001000000 | SOH_TYPE | Analogue and Digital Monitors |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp 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 Readback |
| | ---xxxxx-------- | ADC_MUX | ADC Multiplexer |
| | --x------------- | ADC_WR | ADC WR Line Status |
| | -x-------------- | E2PROM_WEN | $E^2$Prom 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 | Software Boot Code Identifier |
| | --x------------- | HOT_START | Hot Start |
| | -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 | 0000000000000000 | | Digital Monitors 3 − Spare |
| 19 | xxxxxxxxxxxxxxxx | | Digital Monitors 4 |
| | ---------------x | C1_PHASE_1 | Camera 1 Shutter Phase 1 |
| | --------------x- | C1_PHASE_2 | Camera 1 Shutter Phase 2 |
| | -------------x-- | C1_PHASE_3 | Camera 1 Shutter Phase 3 |
| | ------------x--- | C1_PHASE_4 | Camera 1 Shutter Phase 4 |
| | -----------x---- | C1_ON | Camera 1 On (FET control) |
| | ----------x----- | C1_SPARE1 | Camera 1 Spare |
| | ---------x------ | C1_DEI_ON | Camera 1 Deice 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 Readback |
| | --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_1 | Camera 2 Shutter Phase 1 |
| | --------------x- | C2_PHASE_2 | Camera 2 Shutter Phase 2 |
| | -------------x-- | C2_PHASE_3 | Camera 2 Shutter Phase 3 |
| | ------------x--- | C2_PHASE_4 | Camera 2 Shutter Phase 4 |
| | -----------x---- | C2_ON | Camera 2 On (FET control) |
| | ----------x----- | C2_SPARE1 | Camera 2 Spare |
| | ---------x------ | C2_DEI_ON | Camera 2 Deice 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_1 | Camera 3 Shutter Phase 1 |

| | | | |
|---|---|---|---|
| | --------------x- | C3_PHASE_2 | Camera 3 Shutter Phase 2 |
| | -------------x-- | C3_PHASE_3 | Camera 3 Shutter Phase 3 |
| | ------------x--- | C3_PHASE_4 | Camera 3 Shutter Phase 4 |
| | -----------x---- | C3_ON | Camera 3 On |
| | ----------x----- | C3_SPARE1 | Camera 3 Spare |
| | ---------x------ | C3_DEI_ON | Camera 3 Deice 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 Readback |
| | ----0000000----- | | Spare |
| | ---x------------ | 1553_SSFLG | 1553 SSFlag |
| | --x------------- | 1553_INCMD | 1553 In Command |
| | -x-------------- | 1553_MEMEN | 1553 Memory Enable |
| | x--------------- | 1553_RDYD | 1553 ReadyD |
| 23−31 | | | Spare |

### 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 | 0000000010000000 | SOH_TYPE | Flat Field Table Checksums |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp 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 |
| 6 | xxxxxxxxxxxxxxxx | CSUM_BLK3 | Block checksum 3 |
| 7 | xxxxxxxxxxxxxxxx | CSUM_BLK4 | Block checksum 4 |
| 8 | xxxxxxxxxxxxxxxx | CSUM_BLK5 | Block checksum 5 |
| 9 | xxxxxxxxxxxxxxxx | CSUM_BLK6 | Block checksum 6 |
| 10 | xxxxxxxxxxxxxxxx | CSUM_BLK7 | Block checksum 7 |
| 11 | xxxxxxxxxxxxxxxx | CSUM_BLK8 | Block checksum 8 |
| 12 | xxxxxxxxxxxxxxxx | CSUM_BLK9 | Block checksum 9 |
| 13 | xxxxxxxxxxxxxxxx | CSUM_BLK10 | Block checksum 10 |
| 14 | xxxxxxxxxxxxxxxx | CSUM_BLK11 | Block checksum 11 |
| 15 | xxxxxxxxxxxxxxxx | CSUM_BLK12 | Block checksum 12 |
| 16 | xxxxxxxxxxxxxxxx | CSUM_BLK13 | Block checksum 13 |
| 17 | xxxxxxxxxxxxxxxx | CSUM_BLK14 | Block checksum 14 |
| 18 | xxxxxxxxxxxxxxxx | CSUM_BLK15 | Block checksum 15 |
| 19 | xxxxxxxxxxxxxxxx | CSUM_BLK16 | Block checksum 16 |
| 20 | xxxxxxxxxxxxxxxx | CSUM_BLK17 | Block checksum 17 |
| 21 | xxxxxxxxxxxxxxxx | CSUM_BLK18 | Block checksum 18 |
| 22 | xxxxxxxxxxxxxxxx | CSUM_BLK19 | Block checksum 19 |
| 23 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 20 |
| 24 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 21 |
| 25 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 22 |
| 26 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 23 |
| 27 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 24 |
| 28 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 25 |
| 29 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 26 |
| 30 | xxxxxxxxxxxxxxxx | CSUM_BLK20 | Block checksum 27 |
| 31 | xxxxxxxxxxxxxxxx | CSUM_BLK28 | Block checksum 28 |

### 3.1.4 Command Status Return

All 1553 messages processed by SMEI have status information returned in this packet. The table is filled in a cyclic fashion, and a counter identifies the last filled entry.

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | 0000000010000000 | SOH_TYPE | Command Status Return |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3 | xxxxxxxxxxxxxxxx | CMD_TOT | Total number of commands received |
| 4 | xxxxxxxxxxxxxxxx | CMD_REJ | Total number of commands rejected |
| 5 | xxxxxxxxxxxxxxxx | SOH_TOT | Total number of SOH requests serviced |
| 6 | xxxxxxxxxxxxxxxx | SCI_TOT | Total number of science data requests serviced |
| 7 | 0000000000000xxx | 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 |
| . | | | |
| . | | | |
| 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 |

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.

### 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.

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | 0000001000000000 | SOH_TYPE | Single event upset information |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3 | xxxxxxxx−−−−−−−− | TRM_NCERR | Total number of non−correctable upsets recorded |
| 3 | −−−−−−−−xxxxxxxx | TRM_CERR | Total number of correctable upsets recorded |
| 4 | xxxxxxxx | | Spare |
| 4 | −−−−−−−−0000xxxx | TRM_LAST | Number of last entry filled (0−8) |
| 5 | −xxxxxxxxxxxxxxx | TRM0_ADDR | #0 Address |
| | x−−−−−−−−−−−−−−− | TRM0_NC | #0 1 = non−correctable error |
| 6 | xxxxxxxxxxxxxxxx | TRM0_OV | #0 Incorrect Value |
| 7 | xxxxxxxxxxxxxxxx | TRM0_RV | #0 Replacement Value |
| 8 | −xxxxxxxxxxxxxxx | TRM1_ADDR | #1 Address |

| | | | | |
|---|---|---|---|---|
| | x−−−−−−−−−−−−−−− | TRM1_NC | #1 | 1 = non−correctable error |
| 9 | xxxxxxxxxxxxxxxx | TRM1_OV | #1 | Incorrect Value |
| 10 | xxxxxxxxxxxxxxxx | TRM1_RV | #1 | Replacement Value |
| 11 | −xxxxxxxxxxxxxxx | TRM2_ADDR | #2 | Address |
| | x−−−−−−−−−−−−−−− | TRM2_NC | #2 | 1 = non−correctable error |
| 12 | xxxxxxxxxxxxxxxx | TRM2_OV | #2 | Incorrect Value |
| 13 | xxxxxxxxxxxxxxxx | TRM2_RV | #2 | Replacement Value |
| 14 | −xxxxxxxxxxxxxxx | TRM3_ADDR | #3 | Address |
| | x−−−−−−−−−−−−−−− | TRM3_NC | #3 | 1 = non−correctable error |
| 15 | xxxxxxxxxxxxxxxx | TRM3_OV | #3 | Incorrect Value |
| 16 | xxxxxxxxxxxxxxxx | TRM3_RV | #3 | Replacement Value |
| 17 | −xxxxxxxxxxxxxxx | TRM4_ADDR | #4 | Address |
| | x−−−−−−−−−−−−−−− | TRM4_NC | #4 | 1 = non−correctable error |
| 18 | xxxxxxxxxxxxxxxx | TRM4_OV | #4 | Incorrect Value |
| 19 | xxxxxxxxxxxxxxxx | TRM4_RV | #4 | Replacement Value |
| 20 | −xxxxxxxxxxxxxxx | TRM5_ADDR | #5 | Address |
| | x−−−−−−−−−−−−−−− | TRM5_NC | #5 | 1 = non−correctable error |
| 21 | xxxxxxxxxxxxxxxx | TRM5_OV | #5 | Incorrect Value |
| 22 | xxxxxxxxxxxxxxxx | TRM5_RV | #5 | Replacement Value |
| 23 | −xxxxxxxxxxxxxxx | TRM6_ADDR | #6 | Address |
| | x−−−−−−−−−−−−−−− | TRM6_NC | #6 | 1 = non−correctable error |
| 24 | xxxxxxxxxxxxxxxx | TRM6_OV | #6 | Incorrect Value |
| 25 | xxxxxxxxxxxxxxxx | TRM6_RV | #6 | Replacement Value |
| 26 | −xxxxxxxxxxxxxxx | TRM7_ADDR | #7 | Address |
| | x−−−−−−−−−−−−−−− | TRM7_NC | #7 | 1 = non−correctable error |
| 27 | xxxxxxxxxxxxxxxx | TRM7_OV | #7 | Incorrect Value |
| 28 | xxxxxxxxxxxxxxxx | TRM7_RV | #7 | Replacement Value |
| 29 | −xxxxxxxxxxxxxxx | TRM8_ADDR | #8 | Address |
| | x−−−−−−−−−−−−−−− | TRM8_NC | #8 | 1 = non−correctable error |
| 30 | xxxxxxxxxxxxxxxx | TRM8_OV | #8 | Incorrect Value |
| 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 $E^2$Prom 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 | 0000010000000000 | SOH_TYPE | Paged Region Memory Dump |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp 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 | 0000100000000000 | SOH_TYPE | Fixed Memory Dump |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TYPE | Timestamp 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.

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

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | 0001000000000000 | SOH_TYPE | Software Performance Counters |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3−31 | xxxxxxxxxxxxxxxx | | <**TBD**> |

### 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 | 0010000000000000 | SOH_TYPE | Housekeeping Test Pattern |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3−31 | xxxxxxxxxxxxxxxx | HTP_DATA | Each word is the previous HTP_DATA word + 0x10001 |

### 3.1.10 GPS Time and Attitude Parameters

This packet provides time and attitude data for ground processing. Data received by SMEI from the spacecraft containing GPS time and attitude data is made available here.

| Word | Mask | Mnemonic | Parameter |
|---|---|---|---|
| 0 | 0100000000000000 | SOH_TYPE | Time and Attitude Parameters |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp of the last update of this data packet |
| 3 | xxxxxxxxxxxxxxxx | UTC_MSW | ICD Figure C6−3, word 2 (UTC Seconds MSW) |
| 4 | xxxxxxxxxxxxxxxx | UTC_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) |

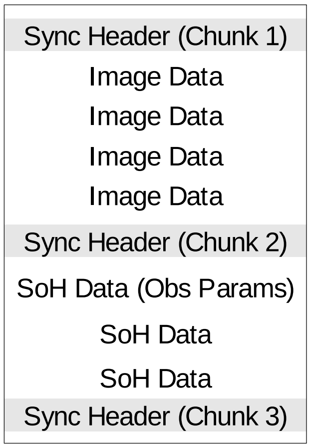
### 3.1.11 Reserved Blocks

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 | x000000000000xxx | SOH_TYPE | Reserved Blocks |
| | 0000000000000001 | | Spare |
| | 0000000000000010 | | Spare |
| | 0000000000000100 | | Spare |
| | 1000000000000000 | | Spare |
| 1 | xxxxxxxxxxxxxxxx | SOH_CRC | Cyclic Redundancy Check |
| 2 | xxxxxxxxxxxxxxxx | SOH_TIME | Timestamp 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 upto 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 (in chunk 1) is followed by a set of housekeeping data in chunk 2. The next chunk 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).

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

Once the calibration parameters are determined, they will be tabulated here.