**Appendix**

**Data-driven Mutation Testing**

**LuxSpace Fault Model**

# 1. Overview

In the case of LXS, data-driven mutation testing is applied to assess the quality of the test cases that exercise the ADCS software interface of the ESAIL system (hereafter, ADCS\_IF\_SW). In ESAIL, the ADCS\_IF\_SW is used to manage and collect data from hardware devices (e.g., sensors). Detailed specifications for the ADCS interface appear in the document *ESAIL-LXS-ICD-P-0184* *ADCS IF SW External ICD.*

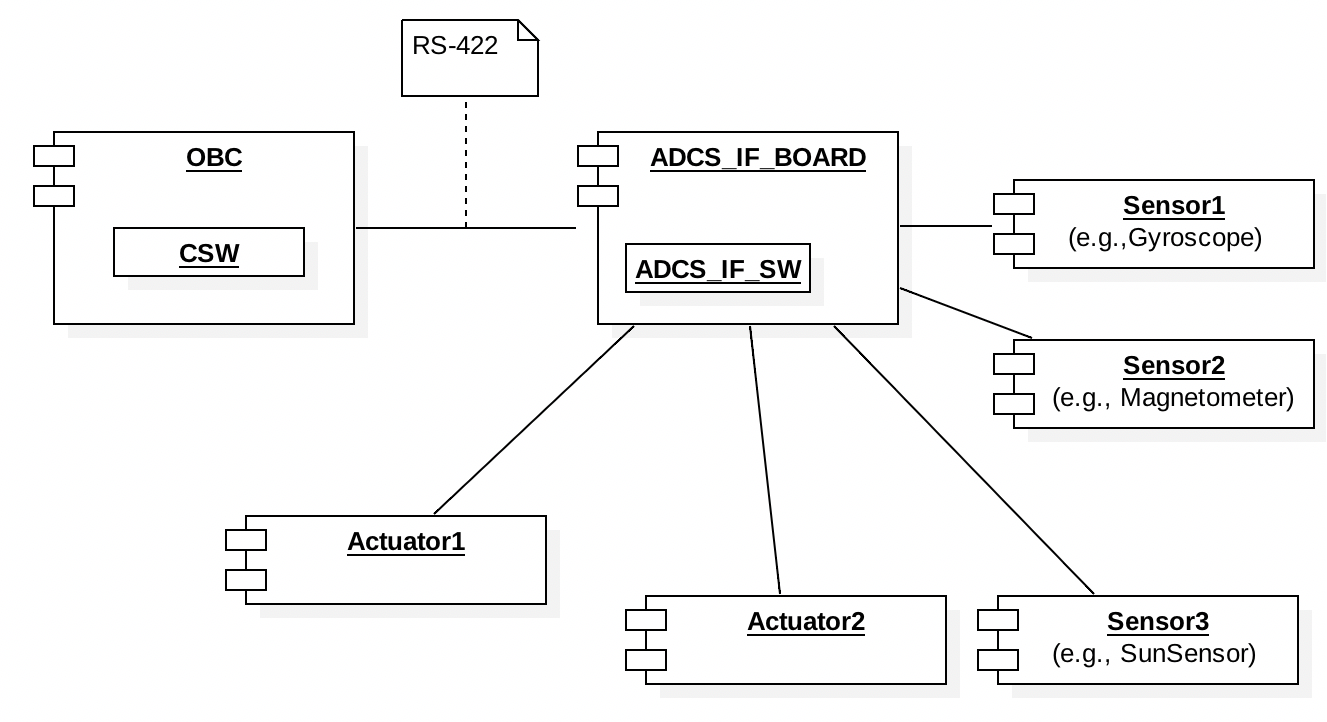


Figure 1: OBC-ADCS integration in ESAIL

Figure 1 provides an overview of the integration between ESAIL OBC and the ADCS board. ESAIL CSW (central software) runs on an onboard controller (OBC) with a Leon 3 microprocessor. The OBC is connected to ADCS interface boards (ADCS\_IF\_BOARD) through RS-422. The ADCS\_IF\_BOARD runs its own controller (ADCS\_IF\_SW). Each board processes data received from sensors and controls actuators. The ADCS\_IF\_SW is the target of data-driven mutation testing and is the software layer where mutation probes are installed.

The ADCS\_IF\_SW implements functions used by the OBC to send data to devices (i.e., set their configuration) and functions that send devices data to the OBC. Since the functions that send data to the devices are typically tested with hardware in the loop, in the context of FAQAS, we will apply data-driven mutation testing only to verify the functions used by the ADCS to send data to the OBC.

ADCS Software

The target for the installation of the mutation probes is the function of the ADCS\_IF\_SW that manages the communication between the ADCS and the OBC, i.e., *ObcRecvBlockCb*. The function is implemented in file *AdcsIf.c*.

The implementation of function *ObcRecvBlockCb* is shown in Figures X-Y. It mainly consists of a switch command (line 138) that generates a response for the OBC by invoking a *data generation method* based on the request received on the data link. For example, Line 146 invokes method *GetIfStatus*, which prepares a response packet containing the information about the ADCS status.

Each *data generation method* receive as input an object of type std::vector, i.e., *newBlock*, that will be used to store the data to be sent to the OBC. The vector *newBlock* acts as a buffer; it contains a number of elements, of type UInt8, that matches the size of the response message indicated in *ESAIL-LXS-ICD-P-0184 (one element per byte).* Table 1 reports, for each feature targeted by data-driven mutation testing, the page in *ESAIL-LXS-ICD-P-0184 that describes the data format, the* ADCS\_IF\_SW function that fill the content of message, and the size of the response message (i.e., the length of the std::vector).

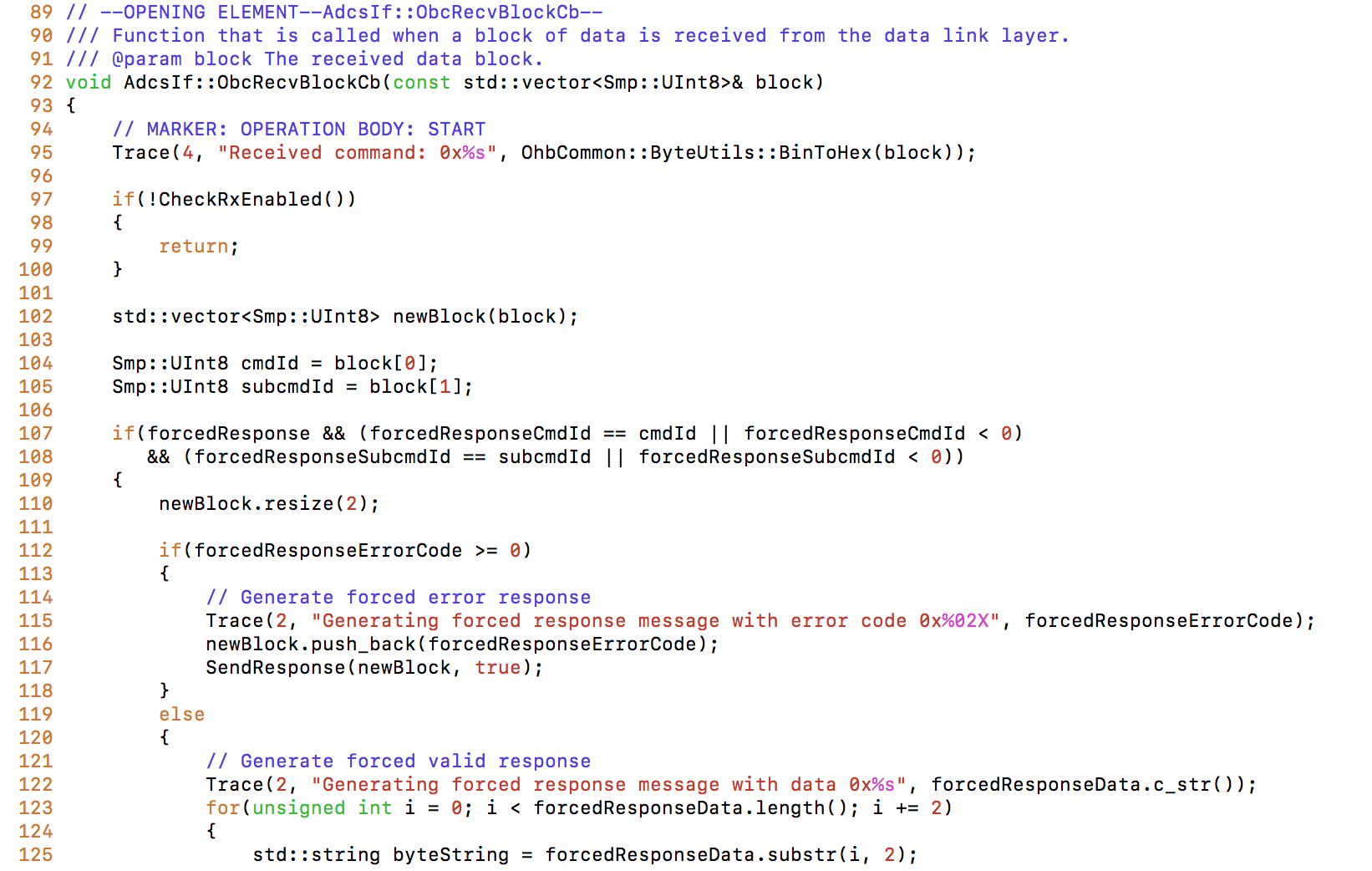
Table 1:Features targeted by data-driven mutation testing and message size

|  |  |  |  |
| --- | --- | --- | --- |
| **ADCS Feature** | **Page** | **ADCS\_IF\_SW function** | **Message size (bytes)** |
| ADCS IF Status | 19 | GetIfStatus | 6 |
| ADCS IF HK | 22 | GetIfHk | 37 |
| GYTM - Gyroscope TM | 34 | GetGyroTm | 21 |
| MMTX - Magnetometer TX | 41 | GetMgtmTm | 2 |
| Sun Sensor TM | 42 | GetSsTm | 48 |
| SSTP - Sun Sensor Temperature | 45 | GetSsTemp | 12 |
| Reaction Wheel TX | 50 | GetRwTm | 2 |
| SpaceCraft HK | 60 | GetIfScHk | 18 |
| Magnetorquer Set PWM RSP | 57 | GetMgtqTm | 39 |

Each invocation of a *data generation method* generates a response (i.e., the newBlock vector) that may either contain the desired result or an error code. The message is sent to the OBC through the invocation of function SendResponse (Lines 298 and 312). When an error code is generated, the data generation method returns CR\_Failure, which is used to trim the buffer before sending back to OBC, this is handled by the parameter *true* passed to SendResponse (Line 312).

...

In Section 3, for each ADCS feature, we report the error codes that might be generated. With data











# 3. Fault Model

## 3.1 ADCS IF Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Byte** | **Bit** | **Description** | **Type** | **Fault type** |
| 1 | 2..0 | **Reset Source**  Provides information about last reset. The bit is cleared after the first read of the status  0 = No reset  1 = Power-on Reset  2 = External Reset (released by JTAG adapter)  3 = Watchdog Reset  4 = Brown-out Reset  5 = JTAG AVR Reset (logic reset by JTAG)  6 = Not used  7 = Not used |  | **NONE** |
| 3 | **ADCS IF ready**  This bit is set when ADCS is ready to read/write to units. In the boot of the ADCS IF shall be a time to initialize all modules and units. After initialization of the ADCS IF, modules and units, shall go to a ready state.  While ADCS IF is not ready, the available commands are:   * ASST * ASHK * ASCT |  | **BF(MIN=0;MAX=0)** |
| 4 | **OBC communication error**  This bit is set if a communication error between OBC and ADCS IF occurred in the last command. The bit is cleared after the first reading of the status  0 = No error  1 = Communication error |  | **BF(MIN=0;MAX=0)** |
| 7..5 | **Unit communication error**  This bit is set if a communication error between ADCS IF and ADCS unit occurred. The bit is cleared after the first read of the status  0 = No error  1 = Communication error |  | **BF(MIN=0;MAX=0)** |
| 2 | 7..0 | **Unit in error**  Provides a list of units in error.  0 = No error  1 = Unit error  Each bit is assigned to one unit:  Bit 0 = Gyroscope unit  Bit 1 = Reaction Wheel  Bit 2 = Magnetorquer  Bit 3 = Magnetometer  Bit 4 = Sun Sensor |  | **BF(MIN=0;MAX=0)** |
| 3 | 7..0 | **Watchdog Reset Counter**  Watchdog Reset counter value.  Increment in every watchdog reset.  Value is stored in non-volatile memory  To clear watchdog reset counter, shall be used the ASCF command. |  | **VAT(T=XXX;D=XXXX)** |
| 4 | 7..0 | **Overall Reset Counter**  Overall reset counter value.  Increment in every device reset.  Value is stored in non-volatile memory  To clear overall reset counter, shall be used the ASCF command. |  | **VAT(T=XXX;D=XXXX)** |
| 5 | 1..0 | **Gyroscope enable**  Enable/Disable status of nominal or redundant bus transceiver.  0 = Disabled both transceivers  1 = Enabled nominal transceiver only  2 = Enabled redundant transceiver only  3 = not existing (reserved for future needs) | **INT** | **INV(MIN=0;MAX=2)** |
| 4..2 | **Reaction Wheel enable**  Enabled/Disabled status of bus transceiver.  0 = Disabled transceiver  1 = Enabled transceiver  7..2 = not existing (reserved for future needs) |  | **BF(MIN=0;MAX=0)** |
| 7..5 | **3 axis Magnetorquer enable**  General Enable/Disable status of the Magnetorquer Driver for all three axis.  0 = Disabled  1 = Enabled  Bit assignement:  Bit 0 = Enabled/Disabled Driver  Bit 1 = 0 not used (reserved for future needs)  Bit 2 = 0 not used (reserved for future needs) |  | **BF(MIN=0;MAX=0)** |
| 6 | 1..0 | **Magnetometer enable**  Enable/Disable status of nominal or redundant bus transceiver.  0 = Disabled both transceivers  1 = Enabled nominal transceiver only  2 = Enabled redundant transceiver only  3 = not existing (reserved for future needs) |  | **INV(MIN=0;MAX=2)** |
| 7..2 | **Sun Sensor board ADC enable**  Enabled/Disabled Sun Sensor board ADC, see Note 3)  0 = Disabled  1 = Enabled  Each bit is assigned to one ADC:  Bit 0 = Enabled/Disabled ADC2  Bit 1 = Enabled/Disabled ADC3  Bit 2 = Enabled/Disabled ADC4  Bit 3 = Enabled/Disabled ADC5  Bit 4 = Enabled/Disabled ADC6  Bit 5 = Enabled/Disabled ADC7 |  | **BF(MIN=0;MAX=5)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x55)**  **IV(VALUE=0x57)**  **IV(VALUE=0x58)**  **IV(VALUE=0x59)**  **IV(VALUE=0x5A)**  **IV(VALUE=0x5B)**  **IV(VALUE=0x5C)** |

## 3.2 ASHK - ADCS IF HK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** | **Type** |  |
| 1 | 7..0 | **VCC1N**  OBC Nominal transceiver circuit voltage |  | **VAT(T=XX;D=XX)** |
| 2 | 7..0 |  |  |
| 3 | 7..0 | **VCC1R**  OBC Redundant transceiver circuit voltage |  | **VAT(T=XX;D=XX)** |
| 4 | 7..0 |  |  |
| 5 | 7..0 | **VCC2**  Gyroscope transceiver/UART circuit voltage |  |  |
| 6 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 7 | 7..0 | **VCC3**  Magnetometer transceiver/UART circuit voltage |  |  |
| 8 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 9 | 7..0 | **VCC4**  Reaction Wheel transceiver/UART circuit voltage |  |  |
| 10 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 11 | 7..0 | **VCCa**  Internal power supply (5.5V), measured with ADC0 |  |  |
| 12 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 13 | 7..0 | **VCCb**  Internal power supply (5.5V), measured with ADC1 |  |  |
| 14 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 15 | 7..0 | **VBUS**  Unit input bus voltage |  | **VAT(T=XX;D=XX)** |
| 16 | 7..0 |  |  |
| 17 | 7..0 | **VCC5**  Supply voltage for ADC2, ADC3, ADC4 and VCCB1.  Sun-sensor PCB |  | **VAT(T=XX;D=XX)** |
| 18 | 7..0 |  |  |
| 19 | 7..0 | **VCC6**  Supply voltage for ADC5, ADC6, ADC7 and VCCB2.  Sun-sensor PCB |  |  |
| 20 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 21 | 7..0 | **VCC5\_IN**  LDO input voltage for ADC2, ADC3, ADC4 and VCCB1.  Sun-sensor PCB |  |  |
| 22 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 23 | 7..0 | **VCC6\_IN**  LDO input voltage for ADC5, ADC6, ADC7 and VCCB2.  Sun-sensor PCB |  |  |
| 24 | 7..0 |  | **VAT(T=XX;D=XX)** |
| 25 | 7..0 | **VCC\_SW1**  SSB internal switched power supply, measured by ADC3  Remark: the voltage VCC\_SW is measured 2 times with two different ADC. This allows to compare the results and conclude for a drift in the ADC’s. |  | **VAT(T=XX;D=XX)** |
| 26 | 7..0 |  |  |
| 27 | 7..0 | **VCC\_SW2**  SSB internal switched power supply, measured by ADC6  Remark: the voltage VCC\_SW is measured 2 times with two different ADC. This allows to compare the results and conclude for a drift in the ADC’s. |  | **VAT(T=XX;D=XX)** |
| 28 | 7..0 |  | **NONE** |
| 29 | 7..0 | **T\_PCB\_TEMP1**  Main Board PCB Temperature, sensor 1  Temperature of VCC DC/DC regulator.  Remark: 1/2 is measured on the same place, it’s to compare the values to discover a measurement failure |  | **VAT(T=XX;D=XX)** |
| 30 | 7..0 |  |  |
| 31 | 7..0 | **T\_PCB\_TEMP2**  Main Board PCB Temperature, sensor 2  Temperature of VCC DC/DC regulator.  Remark: 1/2 is measured on the same place, it’s to compare the values to discover a measurement failure |  |  |
| 32 | 7..0 |  | **VOR(MIN=XX; MAX=XX;D=XX)** |
| 33 | 7..0 | **T\_PCB\_TEMP3a**  Sun Sensor Board PCB Temperature, sensor 3a.  Temperature of VCC5 LDO regulator.  Remark: 3a/b is measured on the same place, it’s to compare the values to discover a measurement failure |  | **VOR(MIN=XX; MAX=XX;D=XX)** |
| 34 | 7..0 |  |  |
| 35 | 7..0 | **T\_PCB\_TEMP3b**  Sun Sensor Board PCB Temperature, sensor 3b.  Temperature of VCC5 LDO regulator.  Remark: 3a/b is measured on the same place, it’s to compare the values to discover a measurement failure |  | **VOR(MIN=XX; MAX=XX;D=XX)** |
| 36 | 7..0 |  |  |
| 37 | 7..0 | **T\_PCB\_TEMP4**  Sun Sensor Board PCB Temperature, sensor 4. Temperature of VCC6 LDO regulator. |  | **VOR(MIN=XX; MAX=XX;D=XX)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x55)**  **IV(VALUE=0x57)**  **IV(VALUE=0x58)**  **IV(VALUE=0x59)**  **IV(VALUE=0x5A)**  **IV(VALUE=0x5B)**  **IV(VALUE=0x5C)** |

## 3.3 GYTM - Gyroscope TM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Unit identifier**  Identification of the unit that addresses the message  0 = Nominal  1 = Redundant | **Int** | **BF(MIN=0,MAX=0)** |
| 21..2 | 7..0 | **Gyroscope Telemetry**  All telemetry data from Gyroscope.  Message is the same sent from Gyroscope unit without adding/removing data | **Hex?** | **NONE?** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)** |

## 3.4 MMTX - Magnetometer TX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Unit identifier**  Identification of the unit that addresses the message  0 = Nominal  1 = Redundant | **Int** | **BF(MIN=0;MAX=0)** |
| 2 | 7..0 | **Self Test Result**  0 = no error  1 = error detected during TX self test  2-255 = reserved | **Hex** | **BF(MIN=0;MAX=0)**  **BF(MIN=1;MAX=7), it simulates reporting a failure for another unit.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)**  **IV(VALUE=0x5D)** |

## 3.5 Sun Sensor TM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Photodiode Q1 current ADC #3** | **Hex** | **BF(MIN=0;MAX=15)?** |
| 2 | 7..0 |  |
| 3 | 7..0 | **Photodiode Q2 current ADC #3** |  |  |
| 4 | 7..0 |  |  |
| 5 | 7..0 | **Photodiode Q3 current ADC #3** |  |  |
| 6 | 7..0 |  |  |
| 7 | 7..0 | **Photodiode Q4 current ADC #3** |  |  |
| 8 | 7..0 |  |  |
| 9 | 7..0 | **Photodiode Q1 current ADC #2** |  |  |
| 10 | 7..0 |  |  |
| 11 | 7..0 | **Photodiode Q2 current ADC #2** |  |  |
| 12 | 7..0 |  |  |
| 13 | 7..0 | **Photodiode Q3 current ADC #2** |  |  |
| 14 | 7..0 |  |  |
| 15 | 7..0 | **Photodiode Q4 current ADC #2** |  |  |
| 16 | 7..0 |  |  |
| 17 | 7..0 | **Photodiode Q1 current ADC #6** |  |  |
| 18 | 7..0 |  |  |
| 19 | 7..0 | **Photodiode Q2 current ADC #6** |  |  |
| 20 | 7..0 |  |  |
| 21 | 7..0 | **Photodiode Q3 current ADC #6** |  |  |
| 22 | 7..0 |  |  |
| 23 | 7..0 | **Photodiode Q4 current ADC #6** |  |  |
| 24 | 7..0 |  |  |
| 25 | 7..0 | **Photodiode Q1 current ADC #5** |  |  |
| 26 | 7..0 |  |  |
| 27 | 7..0 | **Photodiode Q2 current ADC #5** |  |  |
| 28 | 7..0 |  |  |
| 29 | 7..0 | **Photodiode Q3 current ADC #5** |  |  |
| 30 | 7..0 |  |  |
| 31 | 7..0 | **Photodiode Q4 current ADC #5** |  |  |
| 32 | 7..0 |  |  |
| 33 | 7..0 | **Photodiode Q1 current ADC #4** |  |  |
| 34 | 7..0 |  |  |
| 35 | 7..0 | **Photodiode Q2 current ADC #4** |  |  |
| 36 | 7..0 |  |  |
| 37 | 7..0 | **Photodiode Q3 current ADC #4** |  |  |
| 38 | 7..0 |  |  |
| 39 | 7..0 | **Photodiode Q4 current ADC #4** |  |  |
| 40 | 7..0 |  |  |
| 41 | 7..0 | **Photodiode Q1 current ADC #7** |  |  |
| 42 | 7..0 |  |  |
| 43 | 7..0 | **Photodiode Q2 current ADC #7** |  |  |
| 44 | 7..0 |  |  |
| 45 | 7..0 | **Photodiode Q3 current ADC #7** |  |  |
| 46 | 7..0 |  |  |
| 47 | 7..0 | **Photodiode Q4 current ADC #7** |  |  |
| 48 | 7..0 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)** |

## 3.6 SSTP - Sun Sensor Temperature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Temperature reading from ADC #3** | **?** | **VOR(MIN=?;MAX=?;D=?)** |
| 2 | 7..0 |
| 3 | 7..0 | **Temperature reading from ADC #2** |  |  |
| 4 | 7..0 |  |  |
| 5 | 7..0 | **Temperature reading from ADC #6** |  |  |
| 6 | 7..0 |  |  |
| 7 | 7..0 | **Temperature reading from ADC #5** |  |  |
| 8 | 7..0 |  |  |
| 9 | 7..0 | **Temperature reading from ADC #4** |  |  |
| 10 | 7..0 |  |  |
| 11 | 7..0 | **Temperature reading from ADC #7** |  |  |
| 12 | 7..0 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)** |

## 3.7 Reaction Wheel TX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Unit identifier**  Identification of the unit that addresses the message  0 = Nominal  1 = Redundant |  |  |
| 2 | 7..0 | **Self Test Result**  0 = no error  1 = error detected during TX self test  2-255 = reserved |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)**  **IV(VALUE=0x5D)** |

## 3.8 SpaceCraft HK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **TMTC\_SW1**  Identifies the switching position of the TMTC switch 1: the voltage is ~1.1V for position A and 2.2V for position B. 0V or 3.3V will indicate a short or an interruption. | **?** |  |
| 2 | 7..0 |  |  |
| 3 | 7..0 | **TMTC\_SW2**  Identifies the switching position of the TMTC switch 2: the voltage is ~1.1V for position A and 2.2V for position B. 0V or 3.3V will indicate a short or an interruption. |  |  |
| 4 | 7..0 |  |  |
| 5 | 7..0 | **SC\_TEMP1**  Temperature SC-TEMP1 of a sensor in the S/C structure | **?** | **VOR(MIN=x;MAX=X;D=X)** |
| 6 | 7..0 |  |  |
| 7 | 7..0 | **SC\_TEMP2**  Temperature SC-TEMP2 of a sensor in the S/C structure |  |  |
| 8 | 7..0 |  |  |
| 9 | 7..0 | **SC\_TEMP3**  Temperature SC-TEMP3 of a sensor in the S/C structure |  |  |
| 10 | 7..0 |  |  |
| 11 | 7..0 | **SC\_TEMP4**  Temperature SC-TEMP4 of a sensor in the S/C structure |  |  |
| 12 | 7..0 |  |  |
| 13 | 7..0 | **SC\_TEMP5**  Temperature SC-TEMP5 of a sensor in the S/C structure |  |  |
| 14 | 7..0 |  |  |
| 15 | 7..0 | **SC\_TEMP6**  Temperature SC-TEMP6 of a sensor in the S/C structure |  |  |
| 16 | 7..0 |  |  |
| 17 | 7..0 | **SC\_TEMP7**  Temperature SC-TEMP7 of a sensor in the S/C structure |  |  |
| 18 | 7..0 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x56)** |

## 3.9 Magnetorquer Set PWM RSP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Unit identifier Magnetometer**  Identification of the Magnetometer unit that addresses the message  0 = Nominal  1 = Redundant |  | **BF(MIN=0;MAX=0)** |
| 2 | 7..0 | **Magnetometer Data request reply Byte1**  Sync(LSB) (Note 1) |  | **?** |
| 3 | 7..0 | **Magnetometer Data request reply Byte2**  Sync(MSB) (Note 1) |  |  |
| 4 | 7..0 | **Magnetometer Data request reply Byte3**  RAdr (Note 1) |  |  |
| 5 | 7..0 | **Magnetometer Data request reply Byte4**  Sadr (Note 1) |  |  |
| 6 | 7..0 | **Magnetometer Data request reply Byte5**  ReplyMsg (Note 1) |  |  |
| 7 | 7..0 | **Magnetometer Data request reply Byte6**  Bx Low (Note 1) |  |  |
| 8 | 7..0 | **Magnetometer Data request reply Byte7**  Bx Middle |  |  |
| 9 | 7..0 | **Magnetometer Data request reply Byte8**  CS error + Average + pos Clip X + neg Clip X + BX High (Note 1) |  |  |
| 10 | 7..0 | **Magnetometer Data request reply Byte9**  By Low (Note 1) |  |  |
| 11 | 7..0 | **Magnetometer Data request reply Byte10**  By Middle (Note 1) |  |  |
| 12 | 7..0 | **Magnetometer Data request reply Byte11**  spare + pos Clip Y + neg Clip Y + BY High (Note 1) |  |  |
| 13 | 7..0 | **Magnetometer Data request reply Byte12**  Bz Low (Note 1) |  |  |
| 14 | 7..0 | **Magnetometer Data request reply Byte13**  Bz Middle (Note 1) |  |  |
| 15 | 7..0 | **Magnetometer Data request reply Byte14**  spare + pos Clip Z + neg Clip Z + BZ High (Note 1) |  |  |
| 16 | 7..0 | **Magnetometer Data request reply Byte15**  CS (Note 1) |  |  |
| 17 | 7..0 | **Magnetorquer nX Current - on**  Current MTXA\_N when powered |  |  |
| 18 | 7..0 |  |  |
| 19 | 7..0 | **Magnetorquer nX Current - off**  Current MTXA\_N when unpowered |  |  |
| 20 | 7..0 |  |  |
| 21 | 7..0 | **Magnetorquer pX Current - on**  Current MTXA\_P when powered |  |  |
| 22 | 7..0 |  |  |
| 23 | 7..0 | **Magnetorquer pX Current - off**  Current MTXA\_P when unpowered |  |  |
| 24 | 7..0 |  |  |
| 25 | 7..0 | **Magnetorquer nY Current - on**  Current MTYA\_N when powered |  |  |
| 26 | 7..0 |  |  |
| 27 | 7..0 | **Magnetorquer nY Current - off**  Current MTYA\_N when unpowered |  |  |
| 28 | 7..0 |  |  |
| 29 | 7..0 | **Magnetorquer pY Current - on**  Current MTYA\_P when powered |  |  |
| 30 | 7..0 |  |  |
| 31 | 7..0 | **Magnetorquer pY Current - off**  Current MTYA\_P when unpowered |  |  |
| 32 | 7..0 |  |  |
| 33 | 7..0 | **Magnetorquer nZ Current - on**  Current MTZA\_N when powered |  |  |
| 34 | 7..0 |  |  |
| 35 | 7..0 | **Magnetorquer nZ Current - off**  Current MTZA\_N when unpowered |  |  |
| 36 | 7..0 |  |  |
| 37 | 7..0 | **Magnetorquer pZ Current - on**  Current MTZA\_P when powered |  |  |
| 38 | 7..0 |  |  |
| 39 | 7..0 | **Magnetorquer pZ Current - off**  Current MTZA\_P when unpowered |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Bit** | **Description** |  |  |
| 1 | 7..0 | **Error type**  The possible errors are described in the Table 7. | **Hex** | **IV(VALUE=0x51)**  **IV(VALUE=0x52)**  **IV(VALUE=0x53)**  **IV(VALUE=0x54)**  **IV(VALUE=0x56)**  **IV(VALUE=0x5D)**  **IV(VALUE=0x5E)** |