SPIRE

SUBJECT: **Failure Detection Isolation and Recovery Policy in the SPIRE Instrument**

PREPARED BY: **Bruce Swinyard**

DOCUMENT No: **SPIRE-RAL-PRJ-001128**

ISSUE: **Draft for discussion** **Date:** **29 January 2002**
0.1 **1 February 2002**

APPROVED BY: **Ken King** **Date:**

Riccardo Cerruli-Irelli

Distribution

All Subsystem Managers (for internal distribution as appropriate)	
John Delderfield	**RAL**
Ken King	**RAL**
Eric Sawyer	**RAL**
Sergio Molinari	**IFSI**
Sunil Sidher	**RAL**

Change Record

ISSUE	DATE	
First Draft	29 January 2002	New document with discussion of policy and architecture to be used for autonomous failure detection
0.1	1 February 2002	Issued for IBDR and further discussion with OBS team

Table of Contents

Glossary

ADC	Analogue to Digital Converter
BSM	Beam Steering Mirror
CDMS	Command and Data Management System
CDMU	Command and Data Management Unit
CTR	Central Time Reference
DCU	Detector Control Unit
DPU	Digital Processing Unit
DRCU	Detector Readout out and Control Unit
DSP	Digital Signal Processor
FIFO	First In First Out
FMECA	Failure Modes and Effects Criticality Analysis
FPGA	Fused Programmable Gate Array
FPU	Focal Plane Unit
LCL	
MCU	Mechanism Control Unit
S/C	Spacecraft
SCU	Sub-system Control Unit
SMEC	Spectrometer MEChanism
SPIRE	Spectral and Photometric Imaging REceiver

References

Applicable Documents

AD1 SPIRE Data ICD SPIRE-RAL-PRJ-001078
AD1 DPU/DRCU Electrical ICD SPIRE-SAP-PRJ-000451

Reference Documents

RD1 DRCU Specification Document SPIRE-SAP-PRJ-000461

1. SCOPE

This document outlines the Failure Detection, Isolation and Recovery philosophy that will be adopted for the SPIRE instrument in flight. At a later stage it will be expanded to include the detailed analysis of all failure modes identified in the system level FMECA, which has yet to be completed. At this time this document is written in order to inform the design of the instrument on board software architecture and to ensure that all necessary functions will be incorporated into the hardware/software implementation.

The document also covers the instrument responses necessary to failures occurring at the spacecraft level. Again the procedures are identified but not detailed in this document.

2. SYSTEM ARCHITECTURE

2.1 Electronics

Figure 1 shows the electronics system architecture for SPIRE pertinent to the FDIR philosophy (see RD1 for more details). The SPIRE instrument is controlled by a Digital Processing Unit (DPU) that receives command telemetry from the S/C and issues atomic commands (i.e. unique and indivisible instructions) to one of three electronics sub-units that in turn provide the various control signals and power to the Focal Plane Unit (FPU) sub-systems. These units also provide the signal conditioning and data sampling and acquisition electronics. The three sub-units that receive and execute commands from the DPU are as listed here:

- Detector Control Unit (DCU): this provides the bias supplies to the detectors and JFET amplifiers and also has the lock in amplifiers, multiplexers and ADCs for the detector signals.

- The Mechanism Control Unit (MCU) is a three-axis motion controller for the Spectrometer Mechanism (SMEC) and Beam Steering Mirror (BSM). The motion control for both the SMEC and BSM is governed by a Digital Signal Processor (DSP) using fixed software. The DSP also provides limited low-level health checking and sub-system status monitoring.

- The Sub-system Control Unit (SCU) provides the control voltages and signal conditioning for the various FPU thermistors and heaters associated with the temperature monitoring of the FPU; the cooler control and the calibrator control.

Additionally there is a Power Supply Unit (PSU) that is controlled by the SCU and takes the S/C 28V supply voltage directly from the CDMS. This provides the primary power rails to the other electronics units.

Commands are sent from the DPU on one of three bi-directional serial lines to the three sub-units. The sub-units share a common design for the command and data interface based on an FPGA which receives the commands and deals with addressing and command parsing to the individual sub-systems dealt with by the unit. The FPGA also controls the data acquisition from the sub-systems (usually through multiplexed ADCs but via the memory area of the DSP in the case of the MCU) and the sending of data frames to the DPU across a high-speed unidirectional serial link to FIFOs in the DPU.

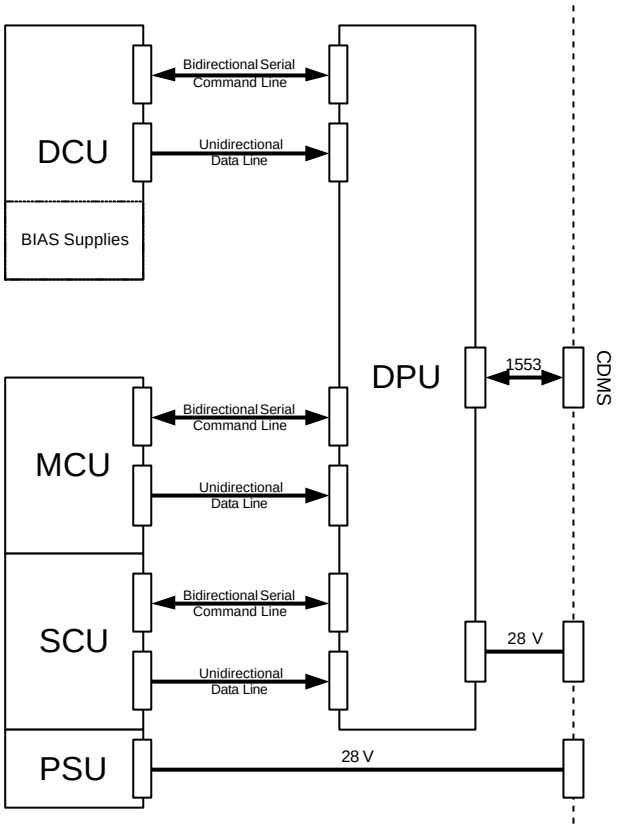

Figure 1: System architecture used to discuss FDIR philosophy

2.2 Command and Data Handling

The S/C CDMS sends instrument commands to the SPIRE DPU at the correct times for the sequencing of the instrument operations. The CDMS is in sole charge of the absolute timing and sequencing of instrument commands to ensure the correct execution of the observations. A time tagged (using on-board time or CTR) command list is stored by the CDMS as an on-board schedule. This schedule is read by the DDMU and commands to the SPIRE instrument are sent to the DPU with 1 sec resolution. The DPU enacts each command as it is received and in the order that they are received.

The commands from the DPU to the SPIRE sub-systems in the DRCU may be generated in one of three basic ways depending on the contents of the telecommand:

1. The telecommand packet references a Command List already stored in the DPU memory. The Command List contains information about the time at which each subsystem command should be sent to the DRCU with a resolution of a few tens of microseconds (using the DPU internal clock)

2. The telecommand packet itself contains a Command List. The list is stored by the DPU and executed in the same way as for 1.

3. The telecommand packet references a DPU function that carries out a given procedure embedded in the on-board software - a control loop algorithm for instance. The DPU will then issue sub-system commands in response to the embedded algorithm as and when

required. Again in principle the timing accuracy available is that of the DPU clock; in practice however these are likely to be run as low level background tasks with a low priority and the actual timing accuracy may be affected by other ongoing operations and will be many tens of microseconds.

3. AUTONOMOUS FAILURE DETECTION

3.1 Overview

The basic philosophy adopted for the autonomous operation of the SPIRE instrument is outlined here:

- Each level of the system architecture is designed so as to protect itself from wrong; inappropriate or interruption to commands from its controlling unit
- Each level shall "report" failures in operation to the level above
- Each controlling level in the system must monitor the levels below it as appropriate and take corrective action in event of detection of a failure.

Figure 2 illustrates the philosophy. At the lowest level in the system are the FPU sub-systems. A failure in one of these is detected by a change in a signal conditioned by the DRCU sub-units. At the next level are the DRCU sub-units. These have limited ability to detect failures in the sub-systems they control.

At the next level is the DPU and on-board software. This is essentially the top level in the SPIRE failure detection system as it monitors all housekeeping values; command execution and data transfer activities from the DRCU and will have the ability to react to any failures detected. All failures detected will be reported to the S/C by an Event Packet – only an "Exception Report" type packet will require action from the CDMS.

At the top level is the CDMS. This will monitor the "Exception Report" type event packets from the DPU and respond according to their contents. It is not expected that the CDMS will directly monitor or react to any instrument housekeeping values. The only exception to this might be the current to the instruments via the 28V lines. This should be monitored by the CDMS and a warning flagged if it reaches some TBD soft limit. It is anyway expected that the LCL will trip automatically if the hard current limit is reached over some defined time period (50 milliseconds?).

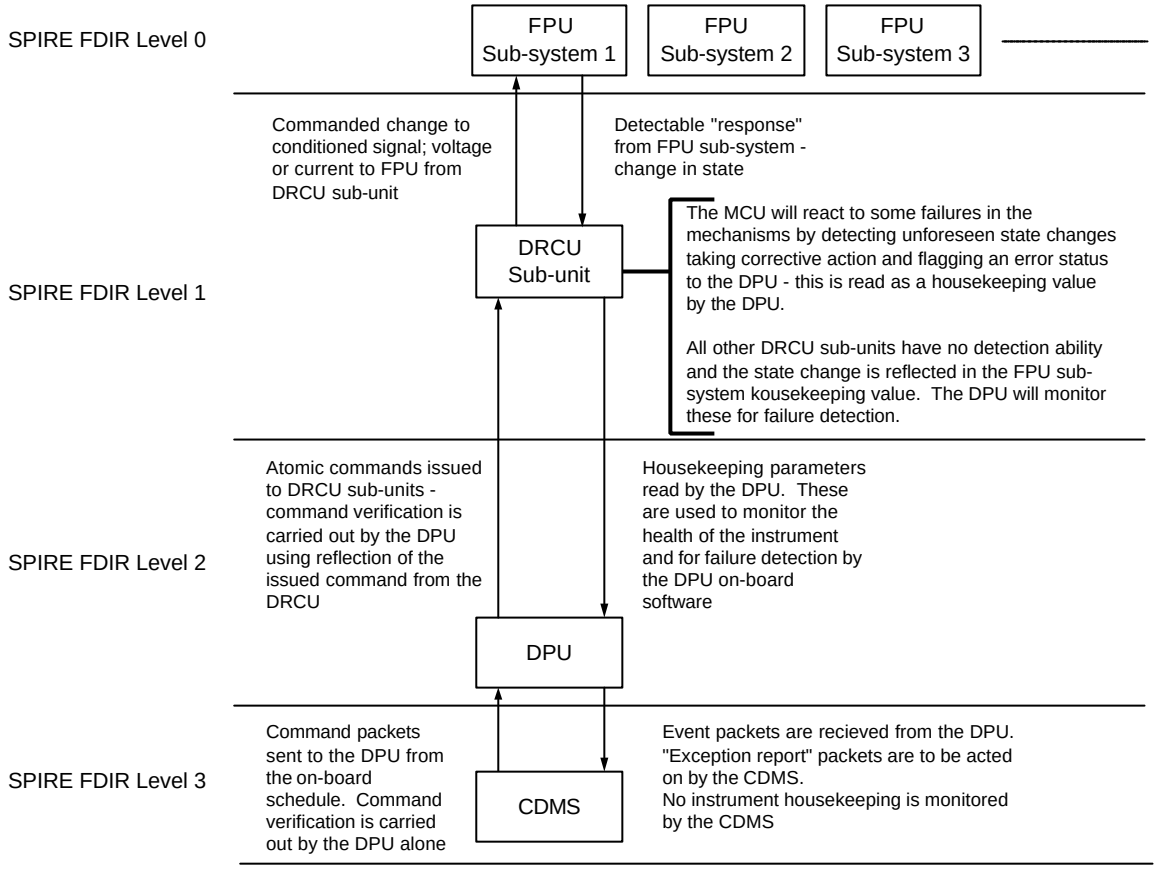

Figure 2: Diagrammatic representation of the SPIRE failure detection philosophy.

3.2 CDMS/DPU Interface, Command and Data Verification

The command structure for SPIRE is described in AD1. The DPU will carry out command verification on the command packets received from the CDMS. The first level verification will be to check that the command is correctly addressed to SPIRE, is of known service type and the command packet is complete using the packet header and checksum information. If these checks are passed then a command reception verification event is sent to the CDMS. It is expected that the CDMS looks at these packets and takes action according to their content.

At the next level of command verification the DPU checks the actaul content of the command packet. Here it verifies each command in the packet to test that it is of known type and there will be some rudimentary check of the parameter value within the command to ensure that carrying out the command will not endanger the instrument. This represents the most basic safety check in the SPIRE autonomy system.

The SPIRE instrument is also required to handle safely any failures in the S/C interface to SPIRE. Failures that might occur are:

- The 1553 bus goes down and no commands are received or data packets taken by the CDMS. The DPU will only execute commands it receives and commands will be packetised so as to be “complete” sequences wherever possible. That is a command packet will not, as far as is possible, leave the instrument in an undefined or unsafe state at

the end of the command sequence execution. Loss of commanding is therefore not necessarily detectable by the DPU. If the SPIRE data packets cannot be sent across the bus then the DPU memory will fill up and a failure will be detected. Appropriate action will be taken by the DPU if this failure is detected. Under these circumstances it is assumed that the CDMS itself will detect the failure and will take corrective action.

- A command packet is lost or becomes out of sequence. The DPU does not "know" what order commands should be sent and will attempt to execute whatever commands it receives irrespective of the state of the instrument. This may mean that inappropriate commands are sent to the electronics sub-units – for instance commanding a sub-system to do something before it is initialised. The instrument will be robust against this type of failure and it should be detected through the normal failure detection when the sub-unit that is commanded cannot respond in the correct manner.

- Loss of synchronisation clock. If the CTR clock pulse from the CDMS is lost the DPU will raise a warning flag and continue. The CTR is not used directly by the DPU for any internal sequencing or control of the instrument.

3.3 DPU/DRCU Command and Data Verification

The commanding protocol from DPU to the electronics sub-units is described in AD2. In general it has two bits for the address; 2 bits for the synchronisation; 8 bits for the command and 20 bits for the command value. On receipt of the command the electronics sub-unit interface FPGA strips the address bits and replaces them with status bits before reflecting the command back to the DPU. The DPU checks the status on receipt – if there are no problems – then the DPU checks bit for bit the reflected command against the command that was sent. This protocol provides the basic command verification system for the instrument control. Procedures will be developed for the response of the DPU to a bad command status or an incorrect bit comparison.

The scientific data are transferred in frames to the DPU via a high speed unidirectional serial line into a FIFO on the DPU. The DPU transfers the contents of the FIFO to the on board memory when the FIFO is half full. In principle the DPU will not inspect the contents of the frames transferred from the sub-systems except to verify that the frame ID is valid; the length of the frame and to perform a checksum. The DPU packetises the data according to the frame ID. There is the facility to packetise whole frames or individual values from within a frame if this proves necessary.

3.4 Electronics Sub-unit Failure Detection

There are a limited number of generic failure types that can occur in the SPIRE sub-systems and associated electronics:

- A Command/Data interface failure causing command receipt corruption – this is detected by the command verification protocol described above. If the address is incorrect the command is never reflected; if the command is corrupted then the status flag will be incorrect and, if the data or command have become corrupted in transmission, then the bit wise comparison of the reflected command will detect this.

- A Command/Data interface failure causing a legitimate command to a sub-system with incorrect value. Here we can envisage that the command was corrupted before transmission in some way – i.e. the DPU thinks it is a correct command so it does not detect a fault. This is dealt with by MCU for some limited circumstances in order to protect the mechanisms from unsafe operation. For all other circumstances the DPU will have to check the HK value pertinent the command set see if the sub-system responded appropriately and to check for unsafe limits.

- Command out of sequence so not appropriate for given state of the sub-system. Again this is dealt with by MCU for some limited circumstances where the command would cause unsafe operation or is not executable. For all other sub-systems it will be seen as an incorrect response of sub-system as detected through the DPU monitoring the HK.

- Correct command but failure in sub-system. Here there are again two cases – dumb units (DCU and SCU) where the failure is detectable by subsequent HK inspection and the MCU which can detect that the command has not been carried out in some limited circumstances and set a status flag in the housekeeping. In both cases the DPU will see the failure through inspection of the HK.

- Failure during normal operation. Again two cases - dumb units only detectable through monitoring HK whilst the MCU can detect a failure, stop the motion and raise status flag in HK. The DPU detects both types through inspection of the HK.

As an example for the last two types of failure: If the SMEC sticks, the MCU will detect an error in the control logic; will halt the SMEC motion autonomously and raise an error status flag in its HK. If a heater goes open circuit a commanded change from the DPU in the heater status will not result in a change in the current monitored by the SCU housekeeping. This however cannot be detected by the SCU itself but only by the on-board software in the DPU, which is charged with monitoring all housekeeping values from the DRCU during normal operations.

3.5 Proposed On-board software failure detection monitoring

The implementation of the on-board software failure detection method and how it should respond to detected failures has not been fully detailed. In this section we propose an outline of how it could be implemented.

Figure 3 illustrates how the autonomous housekeeping verification system might work. The on-board software monitors the DRCU housekeeping at 1 Hz – i.e. every housekeeping parameter is read every second or thereabouts. The values are placed into a stable buffer in memory - how this is done and the precise timing details are TBD. The important point is that there is a list of housekeeping parameter values with known status available for the verification procedure to read.

The procedure that checks the housekeeping reads each parameter in turn from the buffer and checks against a parameter-status configuration which other housekeeping value is associated with this parameter. For example if a current to a heater is to be checked to the configuration will show that the heater status (ON or OFF) should be used to check the parameter limits. The parameter-status configuration also informs the procedure what the actual limits should be for this parameter for a given

sub-system status. We may imagine the configuration as a table as shown in table 1, in reality it may actually be a reference to a stored command list. The requirement on the on-board software is that it shall implement a parameter-status conditional monitoring system.

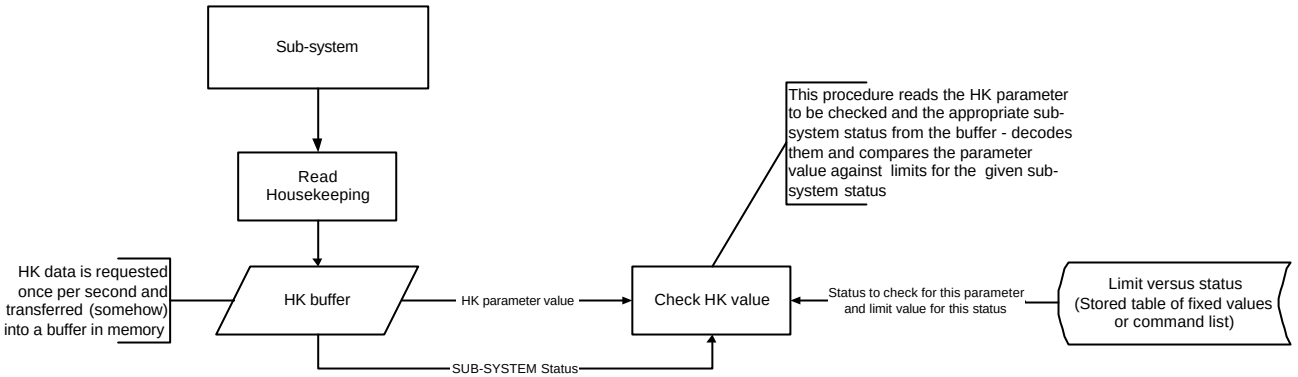

Figure 3: Proposed architecture for autonomous housekeeping checking by the on-board software.

Parameter	Associated Status	Status Condition	Parameter Value limits for this Status
Heater X Current	Heater X PSU Volts	+5.5 (ON) <+0.1 V (OFF)	0 < I < 5 mA I < 0.1 mA

Table 1: Example of parameter to sub-system status configuration. In practice this could be implemented either from a fixed table or by a stored command list.

3.6 SPIRE response to failure detection

We envisage three situations in which a failure detection and associated response may occur:

1. An unexpected response to a command from a sub-system. For example if a change in status is commanded (OFF to ON or change in level) it should be expected that the commanded status and the HK status should agree. If they do not, after some TBD time period, then a failure recovery action is initiated. The action to be taken depends on the situation and each commanding situation will be fully evaluated and the appropriate procedure defined.

2. An inconsistent condition of a housekeeping parameter is detected. This is as discussed in section 3.5. If a housekeeping value is found to be out of (soft) limits for the given status of the sub-system a warning flag is raised through an event packet. The definition of a soft limit is that it shall not initiate failure recovery action, only the raising of a warning flag.

3. An unsafe condition of a housekeeping parameter is detected. If, under any circumstances and irrespective of the sub-system status, a hard out of limits is detected then failure recovery action will be initiated. Again the action to be taken will depend on the sub-system and parameter. Not every HK parameter will have a safety implication, only those that do will have hard out of limits and associated failure recovery actions defined.