Solar B - EIS

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON **Author: K. Al-Janabi**

EIS ICU Software Architectural Design

Document Number: MSSL/SLB-EIS/DD006.01, 30 June 2000

Distribution:

NRL	G Doschek	
	C Korendyke	
	S Myers	
	C Brown	
	K Dere	
	J Mariska	
NAOJ	H Hara	
	T Watanabe	
RAL	J Lang	
	B Kent	
BU	C Castelli	
	S Mahmoud	
	G Simnett	
Mullard Space Science Laboratory	J L Culhane	
	A Smith	
	A James	.
	L Harra	
	A McCalden	
	C McFee	
	R Chaudery	
	P Thomas	
	W Oliver	
	P Coker	
	R Gowen	
	K Al Janabi	
	M Whillock	
SLB-EIS Project Office	A Dibbens	Orig

Author: Date:

Authorised By Date:

Distributed: Date:

CHANGE RECORD

ISSUE	DATE	PAGES CHANGED	COMMENTS
01	30 June 2000	All new	

Table of Contents

1 Applicable References: 3
2 Glossary and Convention: 3
3 Introduction: 4
4 Purpose: 4
5 Modules communication techniques: 5
6 MDP-ICU Data exchange: 5
6.1 TC packets 5
6.2 Status data 5
6.3 Mission data 6
7 EIS ICU Software modules: 6
7.1 The MDP interface: 8
7.2 Status Interface: 9
7.3 Mission Data Buffer: 9
7.4 TC Buffer module: 9
7.5 Task Manager module: 9
7.6 MHC interface: 10
7.7 CAM interface: 11
7.8 PSU interface: 11
7.9 Sequence interpreter module: 11
7.10 Science module: 12
7.11 Health monitor module: 12
7.12 Watchdog module: 12

1 Applicable References:

1 -	NAO/SLB-EIS/SP/MDP001.01	:	MDP-EIS-ICU Electrical Interface
2 -	MSSL/SLB-EIS/SP007-01	:	EIS Science Requirements
3 -	MSSL/SLB-EIS/SP0013.01	:	Solar-B EIS Modes
4 -	Solar-B, EIS-MHC PDR Package		
5 -	Recommended by the J-side, engineering meeting at MSSL on the 23rd May 2000)		

2 Glossary and Convention:

CAM	EIS Camera
CCD	Charge coupled devises
CMD-ID	Command Identifier
CMD_IF	Command Interface
DSP	Digital Signal Processor
EEPROM	Electrically Erasable Programmable Read Only Memory
FPGA	Field Programmable Gate Array
ICU	Instrument Control Unit
MD	Mission Data (science packets)
MD_IF	Mission Data Interface
MDP	Mission Data Processor (Spacecraft)
PROM	Programmable Read Only Memory
RAM	Random Access Memory
ROE	Read Out Electronics
SC_IF	Spacecraft Interface FPGA (Actel)
ST	Status data (instrument Housekeeping)
ST_IF	Status Interface
TC	Telecommand
TM	Telemetry
UART	Universal Asynchronous Receiver Transmitter

3 Introduction:

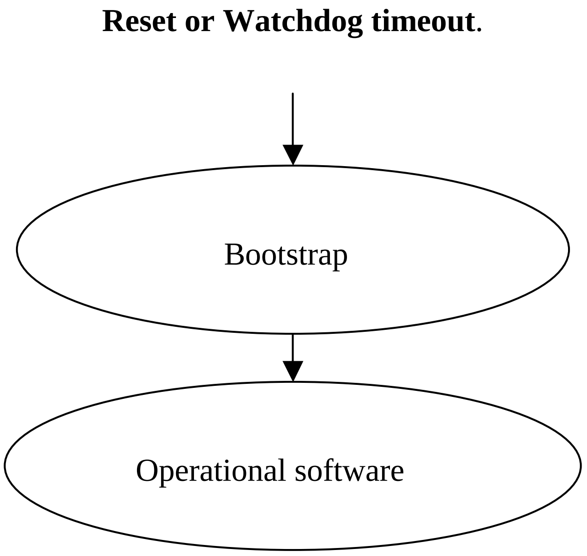
This document describes the Instrument Control Unit (ICU) software structure to be flown on the Solar-B Extreme Ultraviolet Imaging Spectrometer (EIS).

The ICU software shall be written in the C programming language and will use embedded assembly language only when deemed necessary (eg. Interrupt handlers). The software will be run on a DSP-21020 Digital Signal Processor, using Virtuoso Real-Time development tools for embedded systems.

4 Purpose:

The purpose of this document is to identify EIS software modules, their interactions and communication techniques.

Within EIS ICU, there are two independent software components; the Bootstrap and the ICU operational software; as shown below:



EIS Software Schematic

The bootstrap is invoked upon initial powering on of the EIS instrument, a reset or watchdog kick. A minimal “Boot-loader” program is copied from PROM to RAM by a hardware controller, and then the rest of the bootstrap code is copied from PROM to RAM by the boot-loader. The bootstrap program will then copy the application code from EEPROM to RAM and run it (TBC). When the operational software is invoked, Standby mode is entered [3]. At this point the bootstrap ceases to exist and the control is handed over to the Operational software. The Bootstrap will resides in PROM.

5 Modules communication techniques:

In order to synchronise and communicate data between various modules (tasks), the following techniques are used:

Semaphore: A semaphore is a software object, enabling tasks to synchronise their operations. A task signals a semaphore, another detects the semaphore and then performs a specific operation.

Event: Similar to semaphore except it has a faster execution time. Event signalling is the preferred method for communications between low-level software (assembly) (eg. Interrupt handlers) and C structured software.

FIFO: A software object used for transferring data between tasks. The transferred data must be of a fixed size. FIFO techniques are useful for transferring data to software buffers.

Mailbox: A software object suitable for sending data around the system. Unlike FIFOs, a variable data size can be sent using mailboxes.

6 MDP-ICU Data exchange:

The ICU exchanges three types of data with the MDP. These are as follows:

6.1 TC packets

Telecommand packets which consist of a command identifier (one byte) followed by up to 132 bytes, as illustrated below:

CMD-ID	Command Parameters
Command ID	Data Area
8 bits	Max. 132 bytes

6.2 Status data

Status data packets (instrument HK) consist of a 4 byte header followed by up to 2 kbytes of status data. The general format is as follows:

Header Area		Data Area
Data Type	Packet Size	Status Data
8 bits	24 bits	Max. ~2 kbytes

The following are the EIS status allocations:

Status type 1 - ICU status

Header Area		Data Area
Data Type	Packet Size	EIS Status type 1 Data
8 bits	24 bits	100 bytes

Type 2 – Status type 1 + (PSU & CAM status)

Header Area		Data Area	
Data Type	Packet Size	ICU Status type 1 Data	PSU + CAM status
8 bits	24 bits	100 bytes	150 bytes

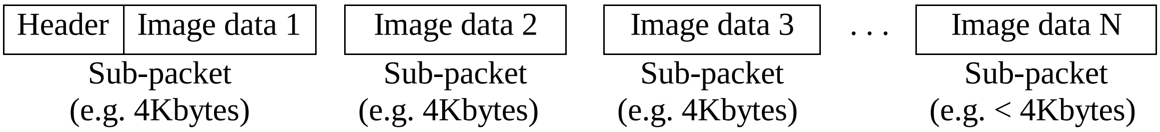
Type 3 – Status type 1 + (MHC status)

Header Area		Data Area	
Data Type	Packet Size	ICU Status type 1 Data	MHC status
8 bits	24 bits	100 bytes	150 bytes

Note that memory dumps packets are also sent via the Status data interface [1].

6.3 Mission data

Science data packets consist of a data header, followed by image data. The maximum size of mission data packet is 256 kpixels (16 bit per pixel). For practical reasons (FIFO hardware size) a mission data packet is sent as a series of 4 kbyte sub-packets, as illustrated below:



7 EIS ICU Software modules:

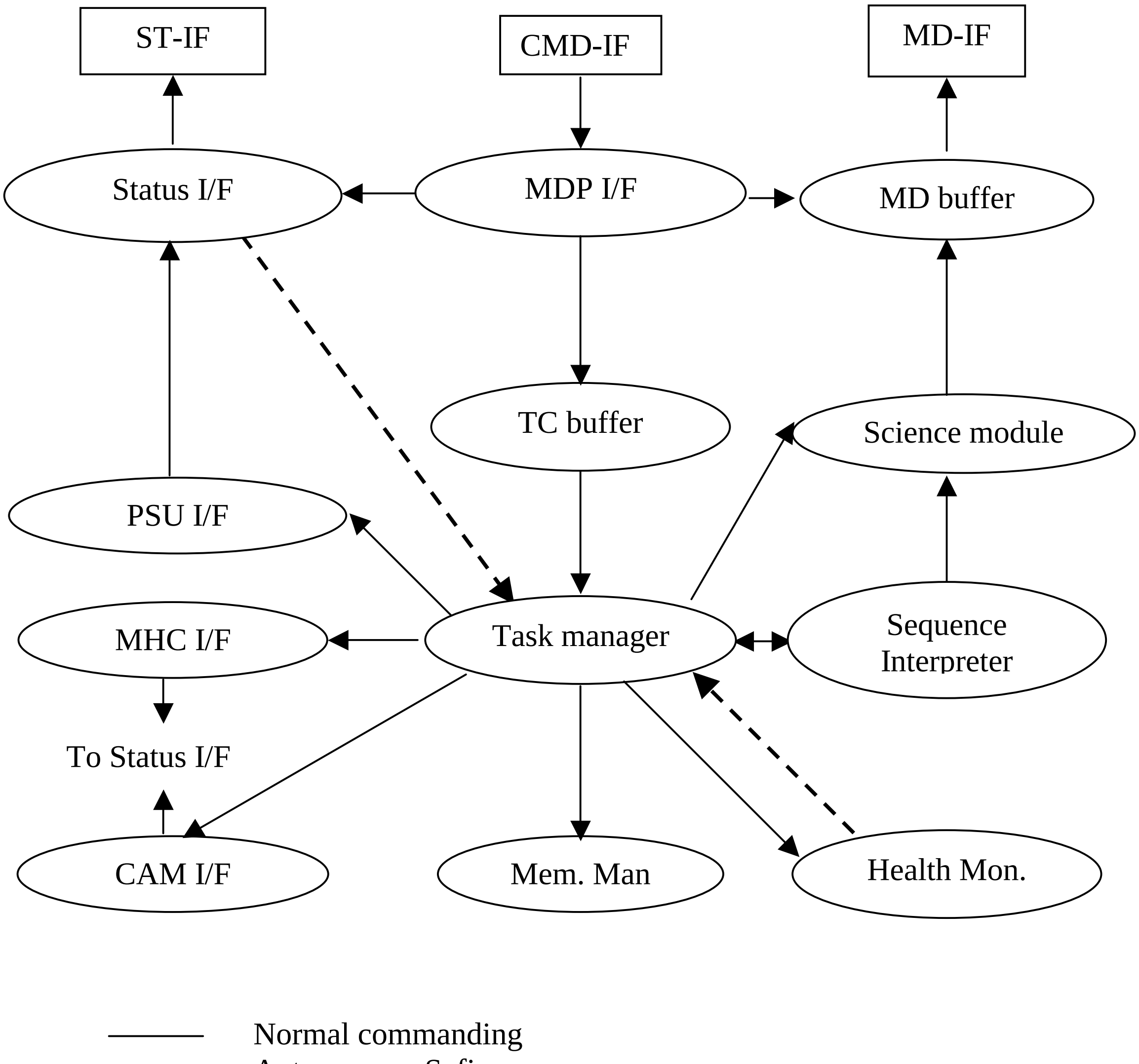
The ICU operational modules and their interfaces are shown below. Within the following text, the following terminology is used:

1) Interface: A software object (module) which interfaces with hardware. Such software fragments synchronise and control hardware operations.

2) Module: A software object which contains one or more tasks.

3) Task: A software object which performs specific operations.

The following is the functional description of the software modules.



The EIS ICU Software Structure

7.1 The MDP interface:

Central to EIS software modules is the MDP software interface. This interface is responsible for the following operations:

1) TM/TC handling
2) Time keeping operations (MDP/ICU time synchronisation)
3) XRT Flare flag acquisition

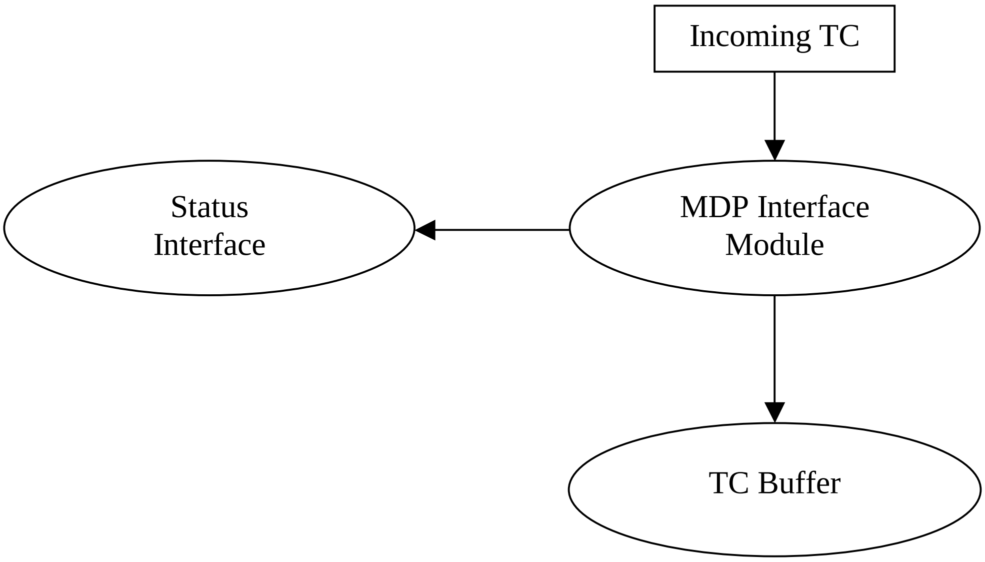
The MDP software interface performs its operations in conjunction with the Spacecraft Interface FPGA – (SC_IF) hardware. The SC_IF hardware issues an interrupt when one of the following operations are performed:

1) A command packet has been written to the EIS hardware command FIFO.

   On Solar-B, there are of two sources for commands sent to instruments; spacecraft ground/time tagged commands and spacecraft autonomous commands.

   Spacecraft autonomous commands (status and memory dump requests) require faster response compared with ground and time tagged commands. Hence they are processed immediately. Ground and time tagged commands are buffered for subsequent processing (the ICU processes buffered commands as fast as possible). These operations are performed as follows:

   Incoming commands are parsed for their CMD-ID. If the commands are status or memory dump requests, then a semaphore is signalled to the Status interface. Any other CMD-ID is sent to the TC buffer, as shown below:



2) A Mission Data (MD) sub-packet has been transmitted to the MDP.

When the spacecraft acquires a MD sub-packet, EIS spacecraft interface generates an interrupt. Upon detecting the interrupt, a semaphore is signalled to the MD buffer so that the next sub-packet can be copied to the ICU MD hardware FIFO.

7.2 Status Interface:

This module handles the processing of the MDP status and memory dump requests [1]. MDP Commands are received via the MDP interface (CMD_IF). When a command is received for one of the three status requests, or a memory dump request is received, the MDP interface signals this module so that the relevant packet is copied to the ST_IF hardware FIFO for transmission to the MDP. Following the status packet being written to the FIFO, this module acquires updated status information of the same type or a memory dump packet for the subsequent request, i.e. always keeps a copy ready for the next request. Failure to supply a packet within a set period (0.5 second) will cause the MDP to generate an error message indicating that the instrument failed to respond to the spacecraft request [1].

Note that this process waits on a semaphore with a 10 second time-out. If no MDP status or memory dump request is received within 10 seconds (MDP software crashed [1]), then this module shall instruct the Task Manager module to terminate science operations and place EIS in a safe state.

7.3 Mission Data Buffer:

This module buffers two mission data packets. When a mission data sub-packet is acquired by the MDP from the MD-IF hardware FIFO, the MDP interface module signals a mission data request to this module and a sub-packet is copied from this buffer to the Mission Data hardware FIFO.

7.4 TC Buffer module:

As stated earlier, ground commands and time tagged commands are buffered by this process for subsequent processing. Note that if incoming commands are processed in a short time then this buffer should remain empty or near empty. However, if incoming TC packets rate exceeds their execution rate, then these commands will be queued (buffered) for subsequent processing.

This buffer is constructed as a software FIFO.

7.5 Task Manager module:

Central to EIS science and engineering operations is the task Manager. This module performs the following operations:

1) Controls EIS Mode transition operations [3].
2) Route commands to their intended destinations.

EIS mode transitions require placing the subsystems in an ON/OFF state. EIS modes perform the following operations:

Standby:

Standby is the first mode entered upon power up or reset. In this mode the Camera and the MHC are OFF.

Manual:

In this mode both the Camera and the MHC are switched ON. In this mode EIS is ready to start Science operations.

Auto:

This mode is invoked when a sequence is run (sequence interpreter operations)

Emergency safe:

This mode is invoked if a fault condition is detected (e.g. Out of range parameter). In this mode only the ICU remains ON.

Bake-out:

In this mode the MHC and Camera are switched OFF. When the mode is verified via EIS ICU status, the heater power is increased in accordance with the required temperature (30 degree C – TBC).

Engineering:

This is a special mode, which maybe used for EIS debugging.

Apart from mode transition operations, the task manager also has the ability of routing commands to subsystems software interfaces and science modules. Incoming commands are routed to other modules, using the CMD-ID as follows:

CMD-IDs	Function
04 - 09	Memory dump (defined by MDP side)
E0 - EF	Memory uplink (defined by MDP side)
20 - 2F	Mode control commands
30 - 3F	PSU commands
40 - 4F	MHC commands
50 - 5F	CAM commands
60 - 6F	Flare operation commands
70 – 7F	Watchdog commands
80 – 8F	Sequence table operations
90 – FF	Spares (excluding E0 – EF)

Note that individual modules are responsible for processing their commands, i.e. the task manager only send commands to their intended destinations.

7.6 MHC interface:

This module controls the access to the MHC hardware resources. The ICU communicates with the MHC via a UART chip, using a byte transfer protocol. The Transmit and Receive functions shall be interlocked such that only one command can be sent to the MHC, at one time [4]. This module performs the following operations:

1) Command transfers to the MHC
2) MHC memory management (uplink and dump)
3) MHC status acquisition. The status is acquired and sent to the status interface when requested.

7.7 CAM interface:

This module controls the access to the CAM hardware resources. The ICU communicates with the CAM via a UART chip, using byte transfer protocol. The Transmit and Receive shall be interlocked such that only one command can be sent to the CAM, at one time. This module performs the following operations:

1) Command transfer to the CAM
2) CAM status acquisition. The status is acquired and sent to the status interface when requested.

7.8 PSU interface:

This module controls the access to the PSU resources. Note that unlike the MHC and the CAM interfaces, the PSU resources are memory mapped and the ICU can directly access the PSU resources. The PSU interface control the following operations:

1) Control the access to the PSU resources (only one command at a time).
2) PSU status acquisitions. The status is acquired and sent to the status interface when requested.

7.9 Sequence interpreter module:

Within EIS, there is a need to perform pre-determined science and engineering operations for a prolonged period. These types of operations are traditionally called sequences. Central to these operations is the sequence interpreter. An engineering sequence is a sequence that contains a finite number of commands that are needed to perform specific operations (e.g. set CCD's bias voltages). A science sequence may contain a finite number of rasters (a typical raster consists of an interleave mirror movements and finite time exposures). It is intended to have a number of default sequences on-board (pre-launch sequences). However sequences can be edited (e.g. change exposure time) or a new sequence can be uplinked. Note that sequence uplinks are treated as memory uplink [1], however, they are treated differently by the on-board software. To ensure the sequence integrity and to guard against misuse (e.g. calling an empty sequence), each sequence is backed by a checksum. The checksum is calculated on-board, at the sequence uplink time.

In order to run a sequence (for the ground or from solar-b deferred command store), the following procedure should be used [5]:

1) From Manual mode, select a sequence number
2) Invoke auto mode.

The sequence then will run (sequence state is reported in the ICU status). A sequence must be either terminated or call another sequence. This should be the last command in the sequence. A sequence can be aborted by either returning to Manual mode or in response to flare flag. Also to ensure the integrity of the science data, a sequence can be paused and resumed. This may be useful for radiation belt operations, as the CCDs background noise level is considered too high for some science operations.

It is anticipated that there will be an on-board storage for 50 sequences, each sequence is 256 bytes deep (TBC). The final number of sequences will be determined by the size of on-board storage size available.

7.10 Science module:

The science process operations are controlled by the sequence interpreter. Following the CCDs read-out completion, the sequence interpreter instructs the science process to perform the required operations. The science module is responsible for the following operations [2]:

1) Mission data extraction (software widowing) and packetisation. Packetised mission data are sent to the mission data buffer.

2) EIS flare flag operations. The exact EIS flare detection algorithm is TBD.

3) Auto exposure operations.

Depending on the final form of EIS science requirements and how complicated the flare detection algorithm under study is going to be, this module may most likely need to be split into more then one process. However, this will have an impact on the ICU software only.

7.11 Health monitor module:

This module is responsible for the EIS instrument health monitor. It performs its operations by checking a predefined set of the ICU and subsystems status (HK) parameters. The parameters to be monitored and the monitoring range are controlled from the ground, via the health monitor table. The health monitor table can be changed (memory uplink command [1]). The health monitor task can be enabled or disable. The status of the health monitor process is reported in the ICU status parameters. When an out of range parameter is detected, this module places EIS in a safe state, via the task manager module. The offending parameter and its value are also reported in the ICU status.

7.12 Watchdog module:

This task is responsible for controlling the watchdog operations. The watchdog can be enabled or disabled (default to enable). When enabled, the watchdog is kicked at regular intervals to prevent it from resetting the ICU. For ground testing (testing the watchdog functionality and the recovery procedure), the ICU can be commanded to stop kicking the watchdog and this will force a watchdog reset.

Note that the watchdog should never be disabled in normal operations.