XMM OPTICAL MONITOR

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON
Author: H.E.Huckle

XMM-OM VXWORKS COMMAND SUMMARY (FM MODEL)

Document: XMM-OM/MSSL/ML/0006.2

Distribution:

XMM-OM Project Office	R Hunt	✓
ESA PX	H Eggel	
CSL	P Rochus	
	S Roose	
Los Alamos National Laboratory	C.Ho	
UCSB	T.Sasseen	
Royal Greenwich Observatory	R Bingham	
University College London	J Fordham	
Mullard Space Science Laboratory	N Bray	✓
	R Card	✓
	M Carter	✓
	R Chaudery	✓
	A Dibbens	
	H Huckle	✓
	H Kawakami	✓
	T Kennedy	✓
	D Self	✓
	P Smith	✓
	P Thomas	✓
	M Whillock	✓
	K Mason	
	A Smith	

Author: Date:

OM Project Office Date:

Distributed: Date:

This copy printed at 03:48 PM on 16-Dec-98

CHANGE RECORD

Issue	Date	Comments
Draft	May‘97	Draft Version for EM Model EMC test
1	Aug ‘97	Issued for EM Delivery
1onc	Nov’97	Updates to FM model begin
2	Dec ‘98	Major update with NCR fixed s/w release.

Index of Commands and Scripts

1. INTRODUCTION

1.1 Conventions Used

- A command typed at the VxWorks prompt is shown in `courier` font.
- Text in *`courier italics`* indicates where numerical values for the indicated parameters should be substituted.
- Hex numbers are indicated, and also typed on the VxWorks console, by preceding the number with a `0x` e.g. the number `16` can be represented as `0x10`.
- In most cases, commands that result in the generation of a telecommand packet begin **`tc_.`**
- **NOTE** - VxWorks commands are case sensitive i.e. `doSomething` is different to `dosomething`.

1.2 SETUP

1. Ensure that the VxWorks crate is connected to its host Sun.
2. Ensure VxWorks Crate and its host Sun are both on.
3. Ensure all relevant hardware is connected e.g. ICU, OM1 etc!
4. Create a terminal window on the Sun
5. From the terminal window, log on to VxWorks i.e. type
   `telnet msslx2`
6. Enable the basic command environment by typing
   `< egse`
7. Enable OBDH polling by typing
   `< startup_icu`

2. BASIC COMMANDS

2.1 Diagnostics

`tc_test`

Sends a test packet to the ICU to check connectivity.

`set_dump_tc` ***`level`***

`level` specifies the dump level for outgoing packet tele-commands.

0 = No dump (default)
1 = Dump Header only
2 = Dump Header and Data.

`set_dump_tm` ***`level`***

`level` specifies the dump level for incoming packet telemetry.

0 = No dump (default)
1 = Dump Header only
2 = Dump Header and Data.

`packet_msg` *`enable_disable`*

This command enables or disables error messages issued when the ‘spacecraft’ polling of the ICU fails. If `enable_disable = 0` or `1` then the message are disabled or enabled respectively.

`rate` ***`rate`***

`rate` specifies the telemetry rate in units of kbits/secs. The allowed values for `rate` are 4,6,8,12,24,100. Any other values are treated as follows: anything bigger than 30 is rounded to 100, otherwise they are rounded to the nearest valid value. If `rate` is omitted, a statement of the current value of `rate` is given and no further action taken. The default value is 24.

2.2 VxWorks Timing Commands

`taskDelay` ***`delay`***

(**Note** uppercase `D`). This command would normally only be inserted in a VxWorks script (see SCRIPTS section). It causes VxWorks to delay the execution of the next command in the script.
NOTE - the units of *`delay`* are 1/60 second.

2.3 TMPSU

2.3.1 Secondary Supplies

`tc_tmpsu_secondaries` ***`on_off`***

Sends a packet to the ICU commanding it to turn on or off the TMPSU secondary supplies.
`on_off` `= 0/1 =` off/on. By default, they are OFF.

`tc_secon`

is ‘shorthand’ for

```
tc_tmpsu_secondaries 1
```

`tc_secoff`

is ‘shorthand’ for

```
tc_tmpsu_secondaries 0
```

2.3.2 Heater Control

tc_heater_config *configuration*

Sends a packet to the ICU commanding it turn to on the heaters as indicated by the configuration number and by the following table. **NB.** Before using this command you must ensure that the standard control algorithm is stopped by sending a `tc_heater_stop` command and the direct heater control task started with the `tc_heater_config_start` command is sent.

***Configuration* (hex)**	**Temperature Control**		**Focussing**	
	HTR1	**HTR2**	**HTR3**	**HTR4**
	Main Interface	**Forward Tube**	**Metering Rods**	**Secondary Mirror**
0xe	On	On	On	-
0xd	On	On	-	On
0xc	On	On	-	-
0xa	On	-	On	-
0x9	On	-	-	On
0x8	On	-	-	-
0x6	-	On	On	-
0x5	-	On	-	On
0x4	-	On	-	-
0x2	-	-	On	-
0x1	-	-	-	On
0	-	-	-	-
All Others	No Change			

tc_heater_config_start

Starts the direct heater control task.

tc_heater_config_stop

Stops the direct heater control task.

`tc_heater_start`

Sends a command packet to the ICU to start the heater control algorithm. This is the default.

`tc_heater_stop`

Sends a command packet to the ICU to stop the heater control algorithm.

`tc_heater_function` ***`fid,param1,param2,param3`***

Sends a command to the ICU to control the heaters. The meaning of the parameters is as follows.

fid	Heater(s)	Algorithm/Function	*param1*	*param2*	*param3*
1	Interface	Closed Loop, Free Running	T_{min}[3]	T_{max}[3]	-
2	Interface	Open Loop, Free Running	On Time[1]	Cycle Time[1]	-
3	Forward	Closed Loop, Synched	T_{min}[3]	T_{max}[3]	Thermistor[2]
4	Forward	Closed Loop, Free Running	T_{min}[3]	T_{max}[3]	Thermistor[2]
5	Forward	Open Loop, Synched	On Time[1]	Cycle Time[1]	-
6	Forward	Open Loop, Free Running	On Time[1]	Cycle Time[1]	-
7	Focusing	Negative, Positive Focusing	On Time[1]	Cycle Time[1]	Focus Direction[4]
8	-	Set Sample Time	Sample Time[5]	-	-

Notes:

1. The ‘On Time’ is the amount the heater spends powered in the ‘Cycle Time’. The times are in units of ‘Sample Time’.
2. A ‘Thermistor’ value of 0 indicates the prime thermistor, a 1 indicates the redundant.
3. T_{min} and T_{max} are in ‘raw’ thermistor units.
4. The value of ‘Focus Direction’ requests focusing direction as follows:
   a) > 0 - positive focusing : only HTR3 (metering rods) powered.
   b) < 0 - negative focusing : only HTR4 (secondary mirror) powered.
   c) 0 - no focusing : both HTR3 and HTR4 are not powered. This is the default.
5. ‘Sample Time’ is in units of seconds in the range 1 through 20. The default is 10 secs.

tc_heater_interface_closed *tmin,tmax*

This is equivalent to a call to `tc_heater_function` with *fid* = 1, except that `tmin` and `tmax` are in degrees centigrade.

tc_heater_interface_open *on_time,cycle_time*

This is equivalent to a call to `tc_heater_function` with *fid* = 2.

tc_heater_forward_closed_synched *tmin,tmax,thermistor*

This is equivalent to a call to `tc_heater_function` with *fid* = 3, except that `tmin` and `tmax` are in degrees centigrade.

tc_heater_forward_closed *tmin,tmax,thermistor*

This is equivalent to a call to `tc_heater_function` with *fid* = 4, except that `tmin` and `tmax` are in degrees centigrade.

tc_heater_forward_open_synched *on_time,cycle_time*

This is equivalent to a call to `tc_heater_function` with *fid* = 5.

tc_heater_forward_open *on_time,cycle_time*

This is equivalent to a call to `tc_heater_function` with *fid* = 6.

tc_heater_focussing *on_time,cycle_time,direction*

This is equivalent to a call to `tc_heater_function` with *fid* = 7.

tc_heater_sample_time *sample_time*

This is equivalent to a call to `tc_heater_function` with *fid* = 8.

`tc_heater_open` *`heater,time_on,cycle_time`*

N.B. EM CODE ONLY.

Sends a packet to the ICU which assigns parameters to the heater control algorithm specifying that heater *`heater`* will be under **open loop** control and that it will be on for 10**`time_on`* secs in a cycle of 10**`cycle_time`* seconds. Note that the heater algorithm is not started by the command, merely that the parameters are assigned (see `tc_heater_start`). The default is that heater 1 is on all the time and heater 2 is on for 160 seconds in a cycle of 200 secs. Heaters 3 and 4 are off.

2.3.3 Filter Wheel Control

2.3.3.1 Low Level

`tc_fw_set_phase` ***`phases`***

Sends a packet to the ICU using the `tc_icb_write` command which energizes the filter wheel motor phases as follows

Phases	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**
0x08	On	-	-	-
0x04	-	On	-	-
0x02	-	-	On	
0x01	-	-	-	On
0	-	-	-	-

NOTES

1) As this command uses the `tc_icb_write` command, you must have previously sent a `tc_icb_start` command.

2) It is possible to combine phases by addition, e.g. the command to turn on both phase1 and phase2 would be

```
tc_fw_set_phase 0x08+0x04
```

By default, all phases are OFF.

2.3.3.2 *Movement*

`tc_fw_move`

Sends a packet to the ICU commanding it to move the filter wheel. The precise way in which the filter wheel is moved is determined by the most recent call to one of

- `tc_fw_filter`
- `tc_fw_to_datum`
- `tc_fw_to_coarse`
- `tc_fw_fine_steps`
- `tc_fw_rel_steps`
- `tc_fw_abs_steps`

These commands are described below.

`tc_fw_stop`

Sends a packet to the ICU commanding it to stop moving the filter wheel. This is only present as a contingency command as the filter should normally stop after the requested movement.

`tc_fw_filter` ***`filter_number`***

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should go to filter *`filter_number`* (range 0 ⇒11). **Note** - this is only obeyed after calls to `tc_fw_to_datum` followed by `tc_fw_move.` The filter numbers have the following meaning.

Filter Number	Filter Wheel Station	Description	Filter Position (steps from datum)
0	0	Blocked	1200
1	1	V	1400
2	2	Magnifier	1600
3	3	U (no bar)	1800
4	4	B	2000
5	5	White	0
6	6	Grism 2 (Visible)	200
7	7	UVW1	400
8	8	UVM2	600
9	9	UVW2	800
10	10	Grism 1 (UV)	1000
11	3	U (Barred)	1837

`tc_fw_to_datum`

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should go to the datum position using both coarse and fine sensors. NOTE - this, and its associated call to `tc_fw_move`, must be called first before `tc_fw_filter/tc_fw_move` is valid

`tc_fw_to_coarse`

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should move to the first positive sensing of the coarse sensor. Note - if it is already at the coarse sensor, it will not move.

`tc_fw_rel_steps` ***`steps`***

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should move `steps` *steps* (range 0 ⇒ `65535 (hex 0xffff))` relative to its current position.

`tc_fw_abs_steps` ***`position`***

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should move it to absolute position *`position`* (range 0 ⇒ `2199)`steps away from the datum position.

`tc_fw_fine_steps` ***`pulses`***

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it should move it a position which is *`pulses`* pulses of fine sensor away from the current position.

`tc_fw_coarse_sensor_current` ***`current`***

Sends a packet to the ICU informing it that, when instructed by `tc_fw_move` to move the filter wheel, the coarse sensor current will be set to *`current`* (range 0 ⇒ 15). The default value is 12*.*

`tc_fw_fine_sensor_current` ***`current`***

Sends a packet to the ICU informing it that, when instructed by `tc_fw_move` to move the filter wheel, the fine sensor current will be set to *`current`* (range 0 ⇒ 15). The default value is 12*.*

`tc_fw_step_rate`
`pull_in_rate,cruise_rate,acceleration`

Sends a packet to the ICU informing it that, when instructed to move the filter wheel by `tc_fw_move`, it will be moved, starting at *`pull_in_rate`* hz, accelerating to *`cruise_rate`* hz at *`acceleration`* hz/sec steps per second.. The default values are 200, 420 and 2000.

2.3.4 Dichroic Control

2.3.4.1 Low Level

`tc_dm_set_phase` ***`phases`***

Sends a packet to the ICU using the `tc_icb_write` command which energizes the dichroic motor phases as follows

Phases	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**
0x08	On	-	-	-
0x04	-	On	-	-
0x02	-	-	On	
0x01	-	-	-	On
0	-	-	-	-

NOTES

1) As this command uses the `tc_icb_write` command, you must have previously sent a `tc_icb_start` command.

2) It is possible to combine phases by addition, e.g. the command to turn on both phase1 and phase2 would be

```
tc_dm_set_phase 0x08+0x04
```

By default, all phases are OFF.

2.3.4.2 Movement

`tc_dm_move`

Sends a packet to the ICU commanding it to move the dichroic motor as requested by `tc_dm_direction.`

`tc_dm_direction` ***`number,method`***

Sends a packet to the ICU informing it that the dichroic, when instructed to move, should do so as follows:

1. If *`method`* is 1, move the dichroic *`number`* steps (range -31 ⇒ 31), a positive value indicating a positive movement.
2. If *`method`* is 0, move the dichroic motor to its maximum excursion in the direction indicated by the sign of *`number`* (zero is considered positive).

A positive movement should be towards the secondary detector.
By default, *`method`* and *`direction`* are zero.

`tc_dm_step_rate` ***`rate.`***

Sends a packet to the ICU informing it that, when moving the dichroic, it will be moved at *`rate`* steps per second (range 1 ⇒ 10). By default, the rate is 2.

2.4 Blue Processing Electronics Commands

2.4.1 Camera Head

`tc_det_camera_head_reset`

Sends a packet to the ICU commanding it to reset the camera head.

`tc_det_camera` ***`enable`***

Sends a packet to the ICU commanding it start (*`enable`* = 1) or stop (*`enable`* = 0) the camera.

2.4.2 High Voltage Control

`tc_det_load_hv_ramp_params` ***`intensifier_voltage,value,ramp_rate,force`***

Specifies that `intensifier_voltage` (1 = cathode, 2 = mcp1, 3 = mcp23) is to be ramped up to value volts at `ramp_rate` (volts/sec) on the next `tc_det_start_hv_ramp` command. If `force` = 1, the ramp will ignore any differences between the requested voltage and HK values (contingency only).

`tc_det_start_hv_ramp`

Initiates the HV ramp sequence defined by the last `tc_det_load_hv_ramp_params`.

`tc_det_stop_hv_ramp`

Stops the current HV ramp sequence.

2.4.3 Window Table Loading

`det_reset_wind`

Resets (clears) the VxWorks internal table of window coordinates. On VxWorks startup, this table is cleared.

`det_wind` ***`x,y,width,height`***

Load the specified window (units CCD pixels) into the VxWorks internal table of window coordinates.

`tc_det_load_windows` ***`verify`***

Constructs and sends two packets that command the ICU to
1. store the coordinates of the required detector windows.
2. load the corresponding bit pattern into the window look-up table.
3. If verify = 1, then the ICU performs a second pass, reading back the data just loaded and comparing it with what was loaded, issuing an exception if they do not match.

The information is obtained from the VxWorks internal table of window coordinates constructed by `det_reset_wind` and `det_wind`

2.4.4 Centroid Lookup Table Loading

`det_equal_bounds`

Set the VxWorks internal table of channel boundaries to 8 equal width ones in the range -1.0 to 1.0 for both axes. On VxWorks startup, this table is empty.

`det_get_bounds <` ***`filename`***

Set the VxWorks internal table of channel boundaries to those specified in the file *`filename`*. The format of the file is the 9 channel boundaries for the x-axis followed by the 9 channel boundaries for the y axis, in both cases in increasing numerical order and free format e.g.

```
-1.0  -0.74  -0.49  -0.23  -0.01  0.27  0.51  0.72  1.00
-1.0  -0.73  -0.48  -0.24   0.01  0.22  0.52  0.76  1.00
```

`tc_det_load_bounds` ***`verify`***

Sends two packets to the ICU commanding it to
1. store the channel boundary information
2. load the centroid lookup table based on the channel boundary information
3. If verify = 1, then the ICU performs a second pass, reading back the data just loaded and comparing it with what was loaded, issuing an exception if they do not match.

The channel boundary information is provided by `equal_bounds` or `get_bounds`.

2.4.5 Exposure Control

`tc_det_start_int`

Sends a packet to the ICU commanding it to start an integration i.e. data is allowed to be written into the FIFO buffers. The start is synchronized to the next occurring end of frame transfer phase of the CCD.

`tc_det_stop_int`

Sends a packet to the ICU commanding it to stop an integration. It stops on the next frame transfer phase and writes to the FIFO are disabled.

`tc_det_acq_mode` ***`acq_mode`***

Sends a packet that informs the ICU of the required detector acquisition mode. There is no default.

acq_mode	Description
0	Low Resolution, Windowed
1	Low Resolution, Full Frame
2	High Resolution, Windowed
3	High Resolution, Full Frame
4	Engineering, X M/N Data
5	Engineering, Y M/N Data
6	Engineering, Event Height
7	Engineering, Event Energy

`tc_det_event_threshold` ***`threshold`***

Sends a packet that informs the ICU of the event detection threshold (range 0 $\Rightarrow$ 255). There is no default. Typical values are 30 for a normal integration and 8 for pulse height distribution building.

`tc_det_enbl_frame_tag` ***`on_off`***

Sends a packet that informs the ICU whether the frame tag is enabled or disabled - *`on_off`* `= 0/1` = off/on. The default is ON. This should be cleared for engineering and the full frame modes.

`tc_det_flood_led_current` ***`current`***

Instructs the ICU to set the flood led current to *`current.`*

2.5 Housekeeping Commands

`tc_hk_start`

This commend sends a packet to the ICU commanding it to start the housekeeping telemetry. By default, it is already started.

`tc_hk_stop`

This commend sends a packet to the ICU commanding it to stop the housekeeping telemetry

2.6 Safing

`tc_safing_start`

Starts the ICU on-board autonomous safing task.

`tc_safing_stop`

Stops the ICU on-board autonomous safing task.

2.7 Watchdog Commands

`tc_watchdog_start`

This commands sends a packet to the ICU commanding to start the watchdog task. By default, it is already started.

`tc_watchdog_stop`

This commands sends a packet to the ICU commanding to stop the watchdog task

`tc_watchdog_ints` ***`timout,reset_int`***

This command sends a packet to the ICU reseting the parameters that control the watchdog. `timout` is in units of 1/256 of a second and specifies the timout loaded into the RBI watchdog register. Its default value is 0xfff, equivalent to 16 secs. `reset_int` is in units of seconds and specifies how often the watchdog task reloads the watchdog register. Its default value is 8 secs.

2.8 Mode Change Commands

`tc_mode` *`mode`*

This command sends a packet to the ICU commanding it to switch to mode `mode`.

Mode	Description
0	Basic
1	Safe
2	Idle
3	Science
4	Engineering

Only the following transitions using this command are allowed - the remaining are rejected by the ICU with a command acceptance failure. A return to Basic can always be obtained using the `reset` command.

From / To	Basic	Safe	Idle	Science	Eng
Basic	Yes	-	-	-	-
Safe	Yes	Yes	Yes	Yes	Yes
Idle	-	Yes	Yes	Yes	Yes
Science	-	-	Yes	Yes	-
Eng	-	-	Yes	-	Yes

2.9 DEMPSU Commands

`tc_dempsu_reset`

Resets the DEMPSU after a latch-up.

`tc_dempsu_on`

Another name for `tc_dempsu_reset`

`tc_dempsu_off`

Turns off the DPU. As this requires a communication path via the SSI to the DPU, the ICU must not be in ‘Basic’ mode.

2.10 Time Management Commands

These commands control the time synchronization of the ICU and DPU to the 'spacecraft' as emulated by the VxWorks EGSE.

`tc_enable_time_synch`

Sends a packet to the ICU informing it that time synchronisation is taking place.

`bcp3`

Sends a BCP3 pulse to the ICU on the next BCP2 pulse (i.e. on a 1 second boundary). This causes a reset of the RBI time counters to zero which then continue to increment.

`tc_add_time_code` ***`coarse,fine`***

Sends a packet to the ICU causing it to add the time specified by coarse and fine, corresponding to the time at the BCP3 pulse, to be added to the still incrementing RBI time registers. `coarse` is the time in seconds, `fine` is the fractional part of the time in units of 2^{-16} of a second.

If the ICU is not in basic mode, a command to synchronize the DPU to the ICU is also sent internally from the ICU to the DPU. It is the equivalent of the command `tc_synch_clk_auto`.

`tc_resync` ***`coarse,fine,delay1,delay2`***

This command
1. sends `tc_enable_time_synch`
2. waits `delay1/60` secs
3. sends `bcp3`
4. waits `delay2/60` secs
5. sends `tc_add_time_code`.

`tc_enable_time_verification`

This command sends a packet to the ICU commanding it to construct and send a time verification packet.

`tc_synch_clk_auto`

Requests that, at the next bcp2/4 pulse (i.e. the next whole number of spacecraft seconds), the seconds field of the DPU clock is set to the least significant 14 bits of the ICU seconds field and the sub-seconds zeroed..

`tc_synch_clk` ***`seconds`***

Requests that, at the next bcp2/4 pulse (i.e. the next whole number of spacecraft seconds), the seconds field of the DPU clock is set to the least significant 14 bits of *`seconds`* and the sub-seconds zeroed..

2.11 Telemetry Management

These commands control those telemetry packets that are specified by SID's (Structure Identifiers). This includes Housekeeping (HK), Non-periodic Housekeeping (NHK) and Science packets.

`tc_enable_sid` ***`sid`***

Enables the packet specified by `SID`.

`tc_disable_sid` ***`sid`***

Disables the packet specified by `SID`.

`tc_disable_all_sid`

Disables all the packets specified by `SIDs`.

`tc_enable_all_sid`

Enables all the packets specified by `SIDs`. **N.B.** This should cause an error message as it is not supported by OM. It is included for completeness.

`tc_report_sid_status`

Report the status of all the packets specified by `SIDs`.

2.12 ICB Commands

These commands cause direct input or output to the Macsbus.

`tc_icb_start`

This command *must* be sent before `tc_icb_write` and `tc_icb_read` commands in order to enable them.

`tc_icb_stop`

This command disables `tc_icb_write` and `tc_icb_read`.

`tc_icb_write` ***`address,subaddress,datum`***

Sends a packet to the ICU commanding it to write *`datum`* to the *`subaddress`* at the supplied ICB *`address`* (see table overleaf)..

`tc_icb_read` ***`address,subaddress`***

Sends a packet to the ICU commanding it to reads the *`subaddress`* at the supplied ICB *`address`*. A Report Task Parameters packet is issued containing the result. A report of the received value is displayed on the console (see table overleaf).

MACSbus Address and Sub-Address Contents

<table>
<tr><th rowspan="2">Addr. (hex)</th><th rowspan="2">Sub System</th><th rowspan="2">Sub-Addr.</th><th rowspan="2">Description</th><th colspan="16">Bits</th></tr>
<tr><th>MSB</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>LSB</th></tr>
<tr><td rowspan="4">7</td><td rowspan="4">TMPSU</td><td>0</td><td>MISC</td><td colspan="3">CS</td><td colspan="4">FW_PSC</td><td>SE</td><td colspan="8">Don’t Care</td></tr>
<tr><td>1</td><td>ADC</td><td colspan="16">Don’t Care</td></tr>
<tr><td>2</td><td>Filter Wheel & Dichroic</td><td>F1</td><td>F2</td><td>F3</td><td>F4</td><td>D1</td><td>D2</td><td>D3</td><td>D4</td><td colspan="8">Don’t Care</td></tr>
<tr><td>4</td><td>Heaters</td><td>H1</td><td>H2</td><td>H3</td><td>H4</td><td colspan="12">Don’t Care</td></tr>
<tr><td rowspan="13">18</td><td rowspan="13">Detector</td><td>2</td><td>Camera Start/Stop</td><td colspan="14">Don’t Care</td><td>R</td><td>SC</td></tr>
<tr><td>3</td><td>Bitmap Address</td><td colspan="8">Bitmap Y Address</td><td colspan="8">Bitmap X Address</td></tr>
<tr><td>4</td><td>Bitmap Datum</td><td colspan="12">Don’t Care</td><td colspan="4">Map Data</td></tr>
<tr><td>5</td><td>Event Detect Threshold</td><td colspan="8">Don’t Care</td><td colspan="8">Event Height Threshold</td></tr>
<tr><td>7</td><td>Table Access Mode</td><td colspan="15">Don’t Care</td><td>TE</td></tr>
<tr><td>8</td><td>Lookup Table Address</td><td colspan="8">Lookup Address (M)</td><td colspan="8">Lookup Address (N)</td></tr>
<tr><td>9</td><td>Lookup Table Datum</td><td colspan="9">Don’t Care</td><td colspan="3">Y sub-pixel</td><td>X</td><td colspan="3">X sub-pixel</td></tr>
<tr><td>10</td><td>Acq Mode, Frame Tag</td><td colspan="12">Don’t Care</td><td>FE</td><td colspan="3">Int Mode</td></tr>
<tr><td>11</td><td>Integration Enable</td><td colspan="15">Don’t Care</td><td>IE</td></tr>
<tr><td>12</td><td>MUX Address</td><td colspan="12">Don’t Care</td><td colspan="4">MUX Address</td></tr>
<tr><td>13</td><td>$V_{cathode}$, V_{mcp23}</td><td>R2</td><td colspan="7">V_{mcp23}</td><td>RV</td><td colspan="7">$V_{cathode}$</td></tr>
<tr><td>14</td><td>Biases</td><td colspan="4">X</td><td colspan="4">X</td><td colspan="4">Fine Sensor Current</td><td colspan="4">Flood LED current</td></tr>
<tr><td>15</td><td>V_{mcp1}</td><td>EH</td><td colspan="5">Don’t Care</td><td>R1</td><td colspan="9">V_{mcp1}</td></tr>
</table>

Key

Abbreviation	Description	PREF
CS	Current Select Secondary Voltages	H890
D1, D2, D3, D4	Phases 1 through 4 of the Dichroic Motor, 1 = energised	H895⇒H898
EH	Enable High Voltages, 1 = Enabled (Alias 9010)	H900
F1, F2, F3, F4	Phase 1 through 4 of the Filter Wheel Motor, 1 = energised	H905⇒H908
FE	Frame Tag enabled if zero (Alias 9280)	H910
FW_PSC	Set current of coarse sensor illuminating LED	H915
IE	Integration Enabled is set to 1 (Alias 9010)	H920
H1, H2	Heaters, 1 = On, H1 = Main Interface, H2 = Forward Tube (Alias 9020)	H925⇒H926
H3, H4	H3 = Metering Rods , H4 = Secondary Mirror (Alias 9020)	H927⇒H928
R	Reset Camera Head (Alias 9290)	H930
R1,R2,RV	Range Bits for V_{mcp1}, V_{mcp23}, $V_{cathode}$ (Alias 9020)	H935⇒H937
SC	1 = Start Frame Readout, 0 = No Frame Readout, bitmap RAM loadable (Alias 9300)	H940
SE	Secondary Voltages, 1 = Enabled (Alias 9010)	H945
TE	Lookup Table access, 0 = Accessed by electronics, table **not** loadable (Alias 9310)	H950
X	Don’t Care	-

Contents of MACSBUS word read back.

MACs Address (hex)	MACs Sub Address	Sub-system	Description	Bits															
				MSB															LSB
7	0	TMPSU	TMPSU Currents and F/W Coarse Sensor	Current Value								Don't Care							CS
18	2	Det-ector	Digital Electronics Status Word	Don't Care								IA	Int Mode			FE	X	TE	ME
	3		Digital Bitmap Datum	Don't Care												Datum			
	4		Digital Centroid Table Datum	Don't Care								0	Y sub-pixel			0	X sub-pixel		
	5		ADC HK, HV Status, Fine Sensor	ADC Address Echo				HE	FP	ADC HK Value									
	6		Initiate ADC	Don't Care															

2.13 Report Task Parameters

`tc_report_task_params` ***`tid,fid`***

Requests a Report Task Parameters packet for the specified *`tid`* and *`fid.`*

2.14 DPU Commands

2.14.1 ICU Functions Controlling the DPU.

This section describes commands affect how the ICU processes the DPU data.

`tc_hbeat_watchdog_start`

Starts the task that monitors the DPU heartbeat. This task will issue an exception if no heartbeat is seen for 25 seconds. It is started by default.

`tc_hbeat_watchdog_stop`

Stops the task that monitors the DPU heartbeat.

`tc_bent_pipe_start`

This places the ICU in ‘bent-pipe’ mode i.e. all blocks are sent straight down to the ‘ground’ (egse) without processing.

`tc_bent_pipe_stop`

This takes the ICU out of ‘bent-pipe’ mode.

The remaining sub-sections describe the DPU commands. In each case ,the specified DPU command(s) is 'wrapped up' in a standard DPU command packet and sent to the ICU. The ICU then 'unwraps' the DPU command and forwards it to the DPU.

2.14.2 Basic DPUOS manipulation and IO commands

`tc_reset_dsp`

Resets the DPU baack to boot idle

`tc_load_dpuos`

Enables DPU operational mode code.

`tc_enbl_dsp` ***`dsp,on_off`***

DSP *`dsp`* is enabled if *`on_off`* = 1. DSP *`dsp`* is disabled if *`on_off`* = 0. *`dsp`* can take the symbolic values of `blue1`, `blue2` or `red`.

`tc_enbl_events` ***`dsp,on_off`***

DSP *`dsp`* is accepts events if *`on_off`* = 1. DSP *`dsp`* rejects events if *`on_off`* = 0. *`dsp`* can take the symbolic values of `blue1`, `blue2` or `red`.

`tc_load_window_constants`

Loads the detector window constants into the blue dsp placed in memory (starting at location 7380) earlier by `tc_prog_det_dtw`.

2.14.3 Memory management

`tc_dump_ram` ***`address,no_dpu_words`***

Causes a dump of *`no_dpu_words`* of ram starting at location *`address`*

`tc_dump_ram_block start_address, number_words, delay`

Dumps *`number_words`* of DPU memory in 1000(hex) blocks starting at address *`start_address`*. *`delay`* specifies, in units of 1/60 sec, how long the command waits before returning the shell script control. This is to allow the DPU time to respond.

`tc_dump_window_ram window, delay`

Dumps *`0x40000`* words of DPU memory starting at *`0x400000 + window * 0x40000`*. *`window`* is in the range 0 to 15. The specified area of memory is where events form window *`window`* (as defined in a given detector event) are accumulated when the DPU is used in an engineering mode. *`delay`* specifies, in units of 1/60 sec, how long the command waits before returning the shell script control. This is to allow the DPU time to respond. It should be at least 12*60.

2.14.4 Exposure configuration commands

`tc_enbl_verbose` *`data_type,enable`*

Enables long form of data, where `enable = 0/1` = disabled/enabled and `data_type` is

`data_type`	Description	Comment
0	tracking	not implemented (but see `tc_report_trk`).
1	field acquisition	not implemented.
2	window parameters	not implemented.
3	reference frame.	Modifies format of DD_REF record
4	Engineering Data	Causes engineering sub-type 7 to be issued with sub-type 3

`tc_set_bpe` *`binning`*

Informs DPU of the status of the "on-chip" binning that the detector will have during the exposure. `binning` = `0/1` = no binning/binning = high/low resolution.

`tc_set_exp_no` *`exp_id`*

Sets the exposure ID to *`exp_id.`*

`tc_set_exposure_time` *`no_frames`*

Sets the exposure time as `no_frames` frames.

`tc_set_frame_time` *`no_cycles`*

Sets the frame time as `no_cycles` cycles (1 cycle = 1/1024 secs).

`tc_set_trk` *`tracking`*

Enable/disables tracking (`tracking = 0/1` = off/on).

`tc_prog_det_wdw` *"`filename`"*

Loads the detector window co-ordinates in file *`filename`* into the memory area starting at 7380.

`tc_prog_mem_wind` *"`filename`"*

Loads the memory window co-ordinates in file *`filename.`*

`tc_prog_ref_stars` *"`filename`"*

Loads the reference star coordinates in file *`filename`*..

`tc_prog_sci_wind` *"`filename`"*

Loads the science window co-ordinates in file *`filename`* .

`tc_report_trk` *`enable`*

Enable/disable the reporting of the tracking history i.e. DA_TRK alerts are/are not sent (`enable/disable = 1/0`)

2.14.5 Major task invocation commands

`tc_init_dpu`

Zeroes the DPU memory used to accumulate images.

`tc_send_command IC_ACQUIRE_FLD`

Commands the DPU to execute the ACQUIRE_FIELD task.

`tc_send_command IC_CHOOSE_GS`

Commands the DPU to execute the CHOOSE_GUIDE_STARS task.

`tc_send_command IC_FLUSH_CMPRS`

Commands the DPU to force compression of the entries in the compression queue.

`tc_send_command IC_INIT_EXP`

Commands the DPU to invoke the INIT_EXP task. The readies the DPU to acquire data in 1k*1k format.

`tc_send_command IC_TRACK_GS`

Commands the DPU to execute the TRACK_GUIDE_STARS task, thus effectively starting the exposure.

`tc_enbl_eng 1,`*`eng_mode`*

Enables the engineering mode given by `eng_mode`.

<table>
<tr><th>eng_mode</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td rowspan="3">Raw data</td><td>with both blue DSPs</td></tr>
<tr><td>1</td><td>with only blue DSP 1</td></tr>
<tr><td>2</td><td>with only blue DSP 2</td></tr>
<tr><td>3</td><td colspan="2">Centroiding data - get channel boundaries and M,N values</td></tr>
<tr><td>4</td><td colspan="2">Full frame, high resolution</td></tr>
<tr><td>5</td><td colspan="2">Centroiding confirmation data</td></tr>
<tr><td>6</td><td colspan="2">Intensifier characteristics - event height information</td></tr>
</table>

2.15 Slot Controller ("Spacecraft") Commands

`clear`

Resets the slot controller

`reset`

Causes a 'cold start' i.e the bootstrap starts, performs safing and then starts up the ICU basic mode.

`resetw`

Causes a 'warm start' i.e the bootstrap starts, copies the rom into ram then awaits a *`start`* command (see below).

`sdt` ***`arg`***

If *`arg = 0x12200aa`*, causes a 'warm start' in which the bootstrap starts, copies the rom into ram then awaits a *`start`* command (see below) - equivalent to `resetw`.

If *`arg = 0x12200bc`*, causes a 'warm start' in which the bootstrap starts, **does not** copy the rom into ram then awaits a *`start`* command (see below).

`start`

Causes the ICU to start basic mode after a warm start.

`raw` ***`address`***

Reads and displays the contents of *`address`* in the data memory of the ICU.

`waw address,value`

Writes *`value`* into *`address`* in the data memory of the ICU.

2.16 Memory Uplink Commands

`packet_load < "filename"`

This command reads the hex file *`filename`* produced by the Tartan Ada linker and constructs and sends the appropriate packets to uplink the code into the ICU.

`tc_packet_load ram_bank < filename`

This commands reads the S-record type file *`filename`* of DPU code and constructs and sends the appropriate packets to uplink the code into the DPU for ram *`ram_bank.`* The bank of ram indicated is the *`0x100000`* words starting at location *`ram_bank*0x100000`*.

2.17 File Archive Commands

`open_packet_archive "filename"`

Opens filename *`filename`* and stores all subsequent telemetry packets in it.

`close_packet_archive`

Closes the packet archive.

2.18 Instrument Station Commands

`enable_is`

This command enables the connection to the instrument station from VxWorks along which telemetry is forwarded for display.

`disable_is`

This command disables the connection to the instrument station from VxWorks along which telemetry is forwarded for display.

3. SCRIPTS

3.1 Introduction

Scripts are simply text files containing a sequence of VxWorks commands. To run e.g. a script contained in the file `commands,` you type

`< commands`

Lines with a # in the very first column are treated as comments and ignored

All scripts are located in **`/home/egse/scripts`**

Script Filename	Description
acquire	Performs and acquire field
bounds	loads a BPE centroid look-up table corresponding to equally spaced boundaries in both x and y.
cgs	Choose guide stars (**not** post acquisition)
cpa	Synonym for `close_packet_archive`
debug	Dumps standard debug memory locations to the screen
eng0	Performs and engineering exposure, sub-mode 0.
eng3	Performs and engineering exposure, sub-mode 3.
eng4	Performs and engineering exposure, sub-mode 4.
eng5	Performs and engineering exposure, sub-mode 5.
eng6	Performs and engineering exposure, sub-mode 6.
exp	Performs single exposure post Choose Guide Stars
flatfield	Performs a flat-field exposure
icu_init	Misnomer - actually enables SID's disabled by default to give extra DPU data-stream diagnostics.
initial_switch_on	Loads up ICU code, switches to safe, loads DPU code.
load_dpu	Loads DPU code
load_icu	Loads ICU code
load_lotsof_dpu	Multiple calls to `load_dpu`
opa	Synonym for `open_packet_archive`
pacgs	Post Acquisition Choose Guide Stars
post_obs	Post observation command sequence
quickstart	Resets ICU, calls `reset_icu`, calls `tc_clean_slate,` calls `bounds`
reset	Resets the ICU
reset_icu	Moves to safe, moves f/w to datum, moves to Idle
tc_clean_slate	Brings up DPU in a clean state, ready to expose.