5.16_V2.1 Monitoring the active flight EEPROM

Last Revised: November 10, 2015
Filename: eeprom_chk

BRIEF FUNCTIONAL DESCRIPTION:

This procedure is for monitoring the EEPROM from the active BEP to check for possible degradation. The EEPROM contains the bootstrap loader and the code that initializes I_CACHE and D_CACHE. Since there is no indication that reading an EEPROM will cause it to degrade faster, its contents can and should be monitored at regular intervals so that errors can be identified and remedial action taken. Once this SOP has been successfully executed, steps 2 to 4 in the below list will be run as part of the command load every six months, and also a few days before any planned BEP reboot, such as when installing new flight software patches.

The `eeprom_cksum` program computes the 32-bit cyclic redundancy checksum of the EEPROM. It is loaded into BEP I_CACHE with a single *writeBep* command, and is executed by an *executeBep* command. The checksum is compared with the 'expected' value; if they match, the program waits for 10 seconds before returning; otherwise it returns immediately, in both cases reporting the actual checksum.

This procedure will run the `eeprom_cksum` program twice; first with the correct checksum for the flight EEPROM, and second with a deliberately bogus checksum. The second bogus run will force an EEPROM dump.

This procedure implements the following basic operations:

1. Confirm the current state of ACIS by verifying BEP HW and SW LEDs, in particular that science is idle and telemetry buffers are clear.

2. Load the `eeprom_cksum` program into I_CACHE

3. Execute the `eeprom_cksum` program with the correct checksum for the flight EEPROM. If the checksum matches (which it should), the program will sleep for 10 seconds.

4. Four seconds after starting step 3, send the command to dump the contents of the EEPROM. This command should fail since the `eeprom_cksum` program is still running. The result code in `pmon` will show an error; nothing is dumped.

5. Repeat step 3 with a deliberately bogus checksum value. In this case, since the checksum doesn't match, it will immediately terminate.

6. Four seconds after starting step 5, send the command to dump the contents of the EEPROM. This time, the command will succeed. Expect a megabyte of telemetry.

ACIS personnel will review the contents of the dump after the procedure has been run.

ASSUMED INSTRUMENT STATE:

This assumes that DPA-A and/or DPA-B is on. ACIS should not be receiving any other concurrent commanding.

SPECIAL INITIAL CONDITIONS:

Spacecraft telemetry should be in Format 2.

OPERATIONAL CONSTRAINTS/CAUTIONS:

The commands in steps 3, 4, and 5 of the table should be built into a Relative Time Sequence (RTS) and loaded into an SCS slot in the OBC for execution. The time sequence of commands is intended to be:

- XBEEPCKSUM (step 3.1)
- a 4 second wait (step 4)
- RBROMDUMP1 (step 5.1)

The crucial timing is the 4 second wait between the XBEEPCKSUM and RBROMDUMP1 commands.

Similarly, the commands in steps 7, 8, and 9 of the table should be built into a Relative Time Sequence (RTS) and loaded into an SCS slot in the OBC for execution. The time sequence of commands is intended to be:

- XBEEPBOGUS (step 7.1)
- a 4 second wait (step 8)
- RBROMDUMP1 (step 9.1)

Again, the crucial timing is the 4 second wait between the XBEEPBOGUS and RBROMDUMP1 commands.

Once the successful EEPROM dump has started, loss of telemetry is OK. Contents of the EEPROM dump can be examined at a later time.

CONTINGENCY PLANS:

REFERENCES:

1. Peter G. Ford memo, April 28, 2015, ”Correcting for ACIS EEPROM corruption (v1.6)”

CHANGE HISTORY:

V0.1
- Initial version, based on memo referenced above.

V0.2
- Incorporates changes to text suggested by Royce Buehler.

V0.3
- Incorporates changes to text suggested by Peter Ford and Paul Plucinsky. Table reformatted by Peter Ford.

V1.0
- Extremely minor edits. Ready for wider review.

V1.1
- Incorporates changes suggested by Peter Ford.

V2.0
- Sent to FOT

V2.1
- Corrected name of program in table.

This page intentionally left blank

Table 1: **ACIS EEPROM corruption check(Page 1)**

Step #	**Title (Revision 5.16_V2.1)**	**Time (mins)**	**Command Description**	**Command Mnemonic**	**Cmd EGSE**	**Seq Key**	**Telemetry Description**	**Telemetry Mnemonic**
1	**Verify current ACIS status**							
1.1	Verify HW LEDs	2					BEP Select BEP Not in Reset BEP FIFO Not Full BEP FIFO Not Empty	1STAT4ST 1STAT5ST 1STAT6ST 1STAT7ST
1.2	Verify SW LEDs	2					BEP is running Science run status Watchdog boot BEP initialization	1STAT0ST 1STAT1ST 1STAT2ST 1STAT3ST
2	**Load eeprom_cksum into I_CACHE**							
2.1	Load eeprom_cksum		writeBep	WBEEPCKSUM				
3	**Execute eeprom_cksum with valid checksum**							
3.1	Execute eeprom_cksum		executeBep	XBEEPCKSUM				
4	**Wait 4 seconds**							
4.1	Wait between commands							
5	**Attempt to dump EEPROM contents**							
5.1	Dump EEPROM		readBep	RBROMDUMP1				
6	**Verify eeprom_cksum status**							
6.1	Verify command status	2						
6.2	Verify command result	2						
7	**Execute eeprom_cksum with bogus checksum**							
7.1	Execute eeprom_cksum		executeBep	XBEEPBOGUS				
8	**Wait 4 seconds**							
8.1	Wait between commands							
9	**Attempt to dump EEPROM contents**							
9.1	Dump EEPROM		readBep	RBROMDUMP1				

Table 1: **ACIS EEPROM corruption check(Page 1)**

Step \#	Expected Value	Units	Telemetry EGSE	Other Verifier	Crit	Description	Notes	RT Con	Tlm Fmt	Min Alt	SIM Pos
1											
1.1	0 or 1 1 1 0				2 2 2 2	0/1 indicates BEP A/B is selected 1 means BEP not in reset 1 means FIFO not full 0 means FIFO empty		Y	2		
1.2	0 or 1 1 1 0				2 2 2 1	bit toggles when BEP running 1 means science idle 1 means no watchdog boot 0 means BEP SW is running		Y	2		
2	**Load eeprom_cksum into I_CACHE**										
2.1			Verify cmdResult ==1 commandEcho 14415					Y	2		
3	**Execute eeprom_cksum with valid checksum**										
3.1			Verify cmdResult==1 commandEcho 14417					Y	2		
4	**Wait 4 seconds**										
4.1	Wait between commands					Commanded 4 second delay					
5	**Attempt to dump EEPROM contents**										
5.1			Verify cmdResult==3 commandEcho 306			Command fails		Y	2		
6	**Verify eeprom_cksum status**										
6.1			bepExecuteReply==0x8e9fdcc0			Correct EEPROM checksum		Y	2		
6.2			EEPROM not dumped								
7	**Execute eeprom_cksum with bogus checksum**										
7.1			Verify cmdResult==1 commandEcho 14427					Y	2		
8	**Wait 4 seconds**										
8.1	Wait between commands					Commanded 4 second delay					
9	**Attempt to dump EEPROM contents**										
9.1			Verify cmdResult==1 commandEcho 306			Command succeeds		Y	2		

Table 1: **ACIS EEPROM corruption check(Page 2)**

Step	**Title**	**Time**	**Command**	**Command**	**Cmd**	**Seq**	**Telemetry**	**Telemetry**
#	**(Revision 5.16_V2.1)**	**(mins)**	**Description**	**Mnemonic**	**EGSE**	**Key**	**Description**	**Mnemonic**
10	**Verify eeprom_cksum status**							
10.1	Verify command status	2						
10.2	Verify command result	2						

Table 1: **ACIS EEPROM corruption check(Page 2)**

Step #	Expected Value	Units	Telemetry EGSE	Other Verifier	Crit	Description	Notes	RT Con	Tlm Fmt	Min Alt	SIM Pos
10	**Verify eeprom_cksum status**										
10.1			bepExecuteReply==0x8e9fdcc0			Correct EEPROM checksum					
10.2			EEPROM dumped								