COS DCE BOOT FSW v1.13 Component Test Results
Requirement 5.1.2.5a HST Error Format

Date: February 13, 2001
Document Number: COS-03-0054
Revision: Initial Release
Contract No.: NAS5-98043
CDRL No.: N/A

Prepared By: Tim Swanson, Software Test Engineer, Design_Net Eng. Date
Reviewed By: K. Brownsberger, COS Sr. Software Scientist, CU/CASA Date
Reviewed By: Grant Blue, COS Software & Operations Manager, BATC Date
Approved By: Barry Welsh, FUV Detector Program Manager, UCB Date
Approved By: John Andrews, COS Experiment Manager, CU/CASA Date

Center for Astrophysics & Space Astronomy
University of Colorado
Campus Box 593
Boulder, Colorado 80309
### REVISIONS

<table>
<thead>
<tr>
<th>Letter</th>
<th>ECO No.</th>
<th>Description</th>
<th>Check</th>
<th>Approved</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Initial Release</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**THE UNIVERSITY OF COLORADO**  
At Boulder  
The Center for Astrophysics and Space Astronomy

**COS DCE BOOT FSW v1.13 Component Test Results**  
Requirement 5.1.2.5a HST Error Format

<table>
<thead>
<tr>
<th>Size</th>
<th>Code Indent No.</th>
<th>Document No.</th>
<th>Rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>COS-03-0054</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Scale: N/A
# Table of Contents

1. Introduction ................................................................................................................. 2  
   1.1 Purpose .................................................................................................................... 2  
   1.2 Scope ....................................................................................................................... 2  
   1.3 Limitations and Constraints .................................................................................... 2  
   1.4 Procedure Overview .............................................................................................. 2  
   1.5 Theory of Test ......................................................................................................... 3  
   1.6 Test Script Implementation ..................................................................................... 4  
      1.6.1 Test Script Arguments ..................................................................................... 4  
      1.6.2 Test Script Coding ........................................................................................... 4  
2. Special Instructions ..................................................................................................... 5  
   2.1 Quality Assurance ................................................................................................... 5  
   2.2 Safety ....................................................................................................................... 5  
      2.2.1 Personal Safety ................................................................................................ 5  
      2.2.2 Test Article and Equipment Safety .................................................................. 5  
   2.3 Contamination ......................................................................................................... 5  
3. Support Requirements ................................................................................................. 5  
   3.1 Personnel ................................................................................................................. 5  
   3.2 Tools, Equipment, and Materials ............................................................................ 6  
   3.3 Data/Software .......................................................................................................... 6  
   3.4 Required Documentation ......................................................................................... 7  
4. Procedure/Task Steps .................................................................................................. 7  
   4.1 Pre-Operation Activities.......................................................................................... 7  
      4.1.1 Make Sure that hks Tools Are Active ............................................................. 7  
      4.1.2 Make Sure that the Proper ROM Is Installed .................................................. 7  
      4.1.3 Log In to the EGSE ......................................................................................... 7  
      4.1.4 Set Current Directory ...................................................................................... 7  
      4.1.5 slogin as eagcos .............................................................................................. 8  
      4.1.6 Set Current Directory ...................................................................................... 8  
      4.1.7 Ensure that Proper Files are Present ................................................................ 8  
   4.2 Operation Execution ............................................................................................... 8  
      4.2.1 Establish Initial Test Conditions ..................................................................... 8  
      4.2.2 Execute the Script ............................................................................................ 9  
   4.3 Post-Operation Activities ....................................................................................... 10  
      4.3.1 Copy Reports to PC Files and Print Them ..................................................... 10  
      4.3.2 Complete The Test Procedure Form ................................................................ 10
1. INTRODUCTION

1.1 PURPOSE

This document presents the Cosmic Origins Spectrograph (COS) Device Control Electronics (DCE) Flight Software (FSW) certification procedure. The purpose of this procedure is to verify that the FSW satisfies Software Requirements according to the method specified in the DCE FSW Test Plan (STP).

1.2 SCOPE

This test procedure comprises the steps necessary to verify that the FSW satisfies Software Requirements Document (SRD) paragraph 5.1.2.5 — HST Error Format.

1.3 LIMITATIONS AND CONSTRAINTS

This test cannot be run in parallel with any other commanding activity directed at the DCE FSW (such as, for example, the periodic transmission of NOOP commands). Test hardware shall be visually inspected, and its configuration noted, prior to conducting this test.

1.4 PROCEDURE OVERVIEW

The procedure requires the hks tools running on the Sun SparcStation Electronic Ground Support Equipment (EGSE) whose network IP address is one of

shorty.ssl.berkeley.edu
taiyo.ssl.berkeley.edu
ginger.ssl.berkeley.edu.

Test time shall be scheduled in advance. The Test Conductor must be logged into the Unix system as user eagcos, and be commanding from the appropriate directory. This directory contains both the test script file and the shell script file; these two files control test execution. The test is conducted by invoking the shell script. This shell script in turn invokes the Perl 5 program UniScript.pl, which resides in its own distinct directory. The test procedure steps have been pre-recorded in the test script file, and are executed interpretively by the UniScript program. The shell script and test script are attached to this document as appendices. As UniScript executes the test script it sends results to the operator console and to two report files, which are also placed in the current directory. After completion of the test script, the Test Conductor can certify successful test execution by examining the contents of the report files and determining that required
outputs are present in them. Printed copies of the report files are attached to the manually completed checklist (Paragraph 4 below) as documentation of the test.

1.5 THEORY OF TEST

When REPORT_DIAGNOSTIC executes it places the LSB of its diagnostic code parameter in the byte-array LFDERR and a corresponding parameter word in LFDERRP at the offset corresponding to that of the entry in LFDERR. The parameter word is obtained by using the diagnostic code (doubled, to index words) as an index into the table HSTERR_TABLE. The word at this location is the desired parameter word. The HSTERR_TABLE contains 34h entries.

This script forces the generation of eight diagnostic codes identified in the COS database as "errors" (as opposed to "warnings"). It then checks the LFDERR and LFDERRP arrays in the HK packet to ensure that (1) the generated diagnostic codes are indeed present in LFDERR in the order in which they were generated and (2) the corresponding "parameter" words have the correct values. A value in LFDERRP is correct if it matches the HK variable per Table 1-1 (which is condensed from the HSTERR_TABLE as read from the FSW code listing).

### Table 1-1: HST Diagnostic Codes

<table>
<thead>
<tr>
<th>Error Code (Hex)</th>
<th>FSW Variable</th>
<th>HK Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Upload Length</td>
<td>LFDCBUF[6]</td>
</tr>
<tr>
<td>02</td>
<td>Data Transfer CRC</td>
<td>LFMXFER</td>
</tr>
<tr>
<td>03</td>
<td>Download Length</td>
<td>LFDCBUF[6]</td>
</tr>
<tr>
<td>04</td>
<td>Serial Number</td>
<td>LFDCBUF[2]</td>
</tr>
<tr>
<td>05</td>
<td>Serial Number</td>
<td>LFDCBUF[2]</td>
</tr>
<tr>
<td>06</td>
<td>Serial Number</td>
<td>LFDCBUF[2]</td>
</tr>
<tr>
<td>11</td>
<td>Serial Number</td>
<td>LFDCBUF[2]</td>
</tr>
<tr>
<td>13</td>
<td>Serial Number</td>
<td>LFDCBUF[2]</td>
</tr>
<tr>
<td>17</td>
<td>Time</td>
<td>LFCTIME</td>
</tr>
<tr>
<td>1B</td>
<td>Time</td>
<td>LFCTIME</td>
</tr>
<tr>
<td>1C</td>
<td>Time</td>
<td>LFCTIME</td>
</tr>
<tr>
<td>1F</td>
<td>Time</td>
<td>LFCTIME</td>
</tr>
<tr>
<td>2F</td>
<td>Opcode</td>
<td>LFDCBUF[0]</td>
</tr>
<tr>
<td>31</td>
<td>MBADISR</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>MBADREGION</td>
<td></td>
</tr>
</tbody>
</table>
1.6  TEST SCRIPT IMPLEMENTATION

1.6.1  Test Script Arguments

The script is parameterized as shown in the following Table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Correct Argument for Version 1.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0</td>
<td>Absolute hex storage address of FSW REPORT_DIAGNOSTIC routine</td>
<td>0B30</td>
</tr>
<tr>
<td>#1</td>
<td>Absolute hex storage address of 8051 “scratch” area for special code</td>
<td>C000</td>
</tr>
</tbody>
</table>

These parameters must be encoded into the shell script u (see Appendix A).

1.6.2  Test Script Coding

The script uses standard UniScript commands and directives. Diagnostic errors are generated by creating calls to the DCE FSW REPORT_DIAGNOSTIC routine — not by trying to induce the erroneous conditions in the 8051 code that would precipitate the diagnostic reporting "naturally" but indirectly. This script creates, as hex data, the following "assembler" code:

C000 7401 MOV A,#DIAG0001 ; generate
C002 1209D0 LCALL REPORT_DIAGNOSTIC ; DIAG00001
C005 7402 MOV A,#DIAG0002 ; generate
C007 1209D0 LCALL REPORT_DIAGNOSTIC ; DIAG0002
C00A 7403 MOV A,#DIAG0003 ; generate
C00C 1209D0 LCALL REPORT_DIAGNOSTIC ; DIAG0003
... ... ...
C023 7408 MOV A,#DIAG001A ; generate
0025 1209D0 CALL REPORT_DIAGNOSTIC ; DIAG001A
; 0028 22 RET ; return to COMMAND task
0029 00 NOP ; gives even byte count

The object code described above is uploaded to the 8051 by means of the LFDUPLOD command. The LFDGOTO command is then used to force the 8051 to LJMP to the just-uploaded code segment. After generating the diagnostics the code executes a RET
instruction that should send control back to the COMMAND processor so the FSW can resume normal operation.

The presence of the HST codes in the LFDERR array and the corresponding "parameter" values can be checked from the HK telemetry.

2. SPECIAL INSTRUCTIONS

2.1 QUALITY ASSURANCE

QA support is required to verify the configuration and setup environment as well as monitoring test steps and verifying results.

2.2 SAFETY

2.2.1 Personal Safety

To ensure the safety of the test personnel during test execution the guidelines contained in Paragraph 3.4 Reference [1] will be adhered to.

2.2.2 Test Article and Equipment Safety

To ensure the safety and well-being of the COS operations bench, SITS, and related test equipment, the following primary safety requirements will be in effect during the execution of this test procedure:

- If access within one (1) meter of COS bench electronics is necessary, wrist straps attached to technical ground shall be used by all personnel involved in handling of any COS test article. Overcurrent and overvoltage shall be set to remove power if nominal limits are exceeded.
- Emergency Power Shutdown — If, during the COS DCE FSW test, power is ON and a severe test equipment failure results in the power system exceeding specified limits, the Test Conductor shall direct or perform shutdown of power.

2.3 CONTAMINATION

All flight hardware shall be handled with clean latex gloves; it shall be covered with clean ESD material and/or stored in a clean flow-bench.

3. SUPPORT REQUIREMENTS

3.1 PERSONNEL
Execution of the COS DCE FSW certification procedure requires the following personnel (to be completed at the Test Readiness Review (TRR):

Test Director: ________________________________
Test Conductor: ________________________________
Test Technician: ________________________________
QA: ________________________________

3.2 TOOLS, EQUIPMENT, AND MATERIALS

The following is a list of tools, equipment, or materials required in this test. Record manufacturer and model, metrology, or property numbers of equipment used, where appropriate. Record calibration due dates where appropriate.

Boot Mode ROM: schematic 27C256

Engineering Ground Support Equipment (see paragraph 1.4). Indicate specific configuration:

<table>
<thead>
<tr>
<th>EGSE</th>
<th>DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>taiyo</td>
<td>ETU</td>
</tr>
<tr>
<td>shorty</td>
<td>DCE #1</td>
</tr>
<tr>
<td>Ginger</td>
<td>DCE #2</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

3.3 DATA/SOFTWARE

The following files must be present:

<p>| Table 3-1: Required Program and Data Files |</p>
<table>
<thead>
<tr>
<th>EGSE (shorty) Directory</th>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\disks\galex\users\galex\tcs\uniscript\</td>
<td>UniScript.pl</td>
<td>UniScript interpreter</td>
</tr>
<tr>
<td>\disks\galex\users\galex\tcs\uniscript\stp5_1_2_5a\</td>
<td>u</td>
<td>Shell script for this procedure</td>
</tr>
<tr>
<td>Ditto</td>
<td>stp5_1_2_5a.tst</td>
<td>Test script for this procedure (Appendix B)</td>
</tr>
</tbody>
</table>

In addition, the hks tools must be active. Directions for activating hks are given in UCB-COS-DOC-1118 (Paragraph 3.4 Reference [4]).
### 3.4 REQUIRED DOCUMENTATION

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document Number</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHB 1700.1(V1-A)</td>
<td>NASA Basic Safety Manual</td>
</tr>
<tr>
<td>2</td>
<td>COS-03-0054</td>
<td>DCE FSW Test Procedure 5.1.2.5a (This document)</td>
</tr>
<tr>
<td>3</td>
<td>UCB-COS-008</td>
<td>COS FUV Detector Software Test Plan</td>
</tr>
<tr>
<td>4</td>
<td>UCB-COS-DOC-1118</td>
<td>COS EGSE Startup Procedure</td>
</tr>
</tbody>
</table>

### 4. PROCEDURE/TASK STEPS

#### 4.1 PRE-OPERATION ACTIVITIES

4.1.1 Make Sure that hks Tools Are Active

Follow the procedure given in Paragraph 3.4 Reference [4].

4.1.2 Make Sure that the Proper ROM Is Installed

Visually verify that the ROM under test is installed: if EEPROM, in U18; if PROM, in U2 and U7.

4.1.3 Log In to the EGSE

In the following steps, the EGSE system (“taiyo”) may be any of the systems listed in Paragraph 1.4. Output, from either the Unix system or from unScript, to the Telnet terminal is represented in typeface. Input from the Test Conductor is represented in **Bold** typeface.

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C:\tcs\us&gt; telnet taiyo.ssl.berkely.edu</td>
<td>Establish connection to taiyo via Telnet client program</td>
</tr>
<tr>
<td>2</td>
<td>Login: xxx Password: --------</td>
<td>Using telnet window, login as user tcs</td>
</tr>
</tbody>
</table>

#### 4.1.4 Set Current Directory

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>tcs@taiyo% cd ~galex/tcs</td>
<td>Change current directory as shown</td>
</tr>
<tr>
<td></td>
<td>tcs@taiyo% pwd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/disks/galex/users/galex/tcs</td>
<td></td>
</tr>
</tbody>
</table>
4.1.5  **slogin as eagcos**

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4    | tcs@taiyo% slogin –l eagcos taiyo.ssl.berkeley.edu  
eagcos@taiyo.ssl.berkeley.edu’s password: *(get from SSL personnel)*  
Last login: Sat Oct 7 10:41:05 2000 from auntem.ssl.berke  
Sun Microsystems Inc. SunOS 5.8 Generic February 2000  
You have mail.  
COS EGSE software version: devel | slogin as eagcos; get password from SSL personnel |

4.1.6  **Set Current Directory**

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
</table>
| 5    | eagcos@taiyo% cd  
/disks/galex/users/galex/tcs/uniscript/stp5_1_2_5a  
eagcos@taiyo% pwd  
/disks/galex/users/galex/tcs/uniscript/stp5_1_2_5a | Change current directory as shown |

4.1.7  **Ensure that Proper Files are Present**

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
</table>
| 6    | eagcos@taiyo% ls –l  
Total 12  
-rw-r--r--  1 tcs eag 1398 Oct 8 18:03  
stp5_1_2_5a.tst  
-rw-r--r--  1 tcs eag 62 Oct 9 17:44 u | List files; the .tst file and the shell script u should be present |

4.2  **OPERATION EXECUTION**

4.2.1  **Establish Initial Test Conditions**

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
</table>
| 7    | eagcos@taiyo% set path=(/path ~dbb/scripts/bin)  
eagcos@taiyo% pkill cosnoopy | Set path as shown to enable access to hks tools; disable automatic generation of LFDNOOPs |
4.2.2 Execute the Script

<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>eagcos:taiy0% <code>sh u</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$pstring=0B30,C000,0,0,0,0,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parameters are: Script File:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stp5_1_2_5a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#0: 0B30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#1: C000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#2: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#3: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#4: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#5: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#6: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#7: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report file</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;/disks/galex/users/galex/tcs/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ver_1_13/stp5_1_2_5a/stp5_1_2_</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5a.rp1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>successfully opened.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report file</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;/disks/galex/users/galex/tcs/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ver_1_13/stp5_1_2_5a/stp5_1_2_</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5a.rp2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>successfully opened.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Script file</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/disks/galex/users/galex/tcs/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ver_1_13/stp5_1_2_5a/stp5_1_2_</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5a.tst</td>
<td></td>
</tr>
<tr>
<td></td>
<td>successfully opened at level 0.</td>
<td></td>
</tr>
</tbody>
</table>

"Address of REPORT_DIAGNOSTIC routine is 0xB30."
"Address of test code is 0xC000."
"Are these addresses correct? If so, press Y;"
"If incorrect press N and edit shell for stp5_1_2_5a.tst." y Continuing.
"Sending two PORs followed by one-second WAITs"
"Sending LFDDIAGC; transmitting diagnostic-generation code."
<table>
<thead>
<tr>
<th>Step</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LFDDIAGC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 0: HKV0=1; HKV1=0; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=7; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=1; wc=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;(2) Sending LFDUPLOD&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LFDUPLOD PROG,L1,CRC1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 0: HKV0=4; HKV1=2; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=3; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=4; wc=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;Sending LFDGOTO 0xC000.&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LFDGOTO PROG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 0: HKV0=7; HKV1=5; wc=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=6; wc=4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT 1: HKV1=7; wc=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;Test 5_1_2_5a completed successfully.&quot;</td>
<td>Re-enable generation of LFDNOOPs</td>
</tr>
</tbody>
</table>

4.3 POST-OPERATION ACTIVITIES

4.3.1 Copy Reports to PC Files and Print Them

Using an FTP client, copy the u, stp5_1_2_5a.tst, stp5_1_2_5a.rp1, and stp5_1_2_5a.rp2 files to appropriate PC files. Include these files as Appendices A, B, C, and D with this completed form.

4.3.2 Complete The Test Procedure Form

Ensure that all blank fields in this report are completed correctly and submit the completed report to QA.
<table>
<thead>
<tr>
<th>OPERATION TITLE:</th>
<th>WOA#</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST ARTICLES IDENTIFICATION (including serial and/or part numbers):</td>
<td></td>
</tr>
<tr>
<td>TASKS/STEPS COMPLETED:</td>
<td></td>
</tr>
<tr>
<td>LOCATION:</td>
<td></td>
</tr>
<tr>
<td>TEST STARTED:</td>
<td>TEST TERMINATED</td>
</tr>
<tr>
<td>TIME: ______ Hr/Min</td>
<td>TIME: ______ Hr/Min</td>
</tr>
<tr>
<td>DATE: ______</td>
<td>DATE: ______</td>
</tr>
<tr>
<td>LOGS USED:</td>
<td></td>
</tr>
<tr>
<td>ANOMALY REPORTS GENERATED:</td>
<td></td>
</tr>
<tr>
<td>COMMENTS:</td>
<td></td>
</tr>
</tbody>
</table>

TEST CONDUCTOR: __________________
Signature/Date

QA REPRESENTATIVE: __________________
Signature/Date
Appendix A. Shell Script u

#!/bin/sh
pkill cosnoopy
perl ../UniScript.pl stp5_1_2_5a "0B30,C000,0,0,0,0,0,0"
cosnoopy&
Appendix B. Test Script stp5_1_2_5a.tst

;*********************************************************************
;*********************************************************************
;  ***                                                                 
; ** BBBB OOO OOO TTTTT OOO N N L Y Y **                           
; ** BBBB OOO OOO OOO N N L Y Y **                                 
; ** BBBB OOO OOO OOO T O O N N N L Y Y **                         
; ** B B O O O O T O O N N N L Y Y **                              
; ** B B O O O O T O O N N N L Y Y **                              
; ** B B O O O O T O O N N N L Y Y **                              
; ** B B O O O O T O O N N N L Y Y **                              
; ** B B O O O O T O O N N N L Y Y **                              
; ** B B O O O O T O O N N N L Y Y **                              
; ** BBBB OOO OOO T OOO N N LLLLL Y **                             
;*********************************************************************
;*********************************************************************

; **********************************************************************
; * Verify Requirement 5.1.2.5a - HST Error Format *
; * * Generate a known pattern of eight HST error codes and supporting *
; * parametric data                                                   *
; * * Verify that the eight error codes and supporting data appear in *
; * the Housekeeping Data                                             *
; * * T H E O R Y :                                                    *
; * * When REPORT_DIAGNOSTIC executes it places the LSB of its *
; * diagnostic code parameter in the byte-array tLFDERR and a *
; * corresponding parameter word in tLFDERRP at the offset *
; * corresponding to that of the entry in tLFDERR. The parameter *
; * word is obtained by using the diagnostic code (doubled, to index *
; * words) as an index into the table HSTERR_TABLE. The word at this *
; * location is then taken as the address of another "table"; at *
; * offset 0 in this table is the desired parameter word. The *
; * HSTERR_TABLE contains 80h entries but only the first 2Fh of these *
; * are used at present, because DIAG002F is the highest diagnostic *
; * code defined.                                                     *
; * * This script forces the generation of eight diagnostic codes *
; * identified in the COS database as "errors" (as opposed to *
; * "warnings"). It then checks the LFDFRR and LFDFRRP arrays in the *
; * HK packet to ensure that (1) the generated diagnostic codes are *
; * indeed present in LFDFRR in the order in which they were *
; * generated and (2) the corresponding "parameter" words have the *
; * correct values. A value in LFDFRRP is correct if it matches the *
; * HK variable per the following table (which is condensed from the *
; * HSTERR_TABLE as read from the FSW code listing). *
; * * Error FSW Variable HK Mnemonic                                   *
; * * *------------------------------------------------------------------
; * * 01 RS17_PAR3 LFDCBUF[6]                                          *
; * 02 RS17_PAR2 LFDCBUF[8]                                           *
; * 03 RS17_PAR1 LFDCBUF[6]                                           *
; * 04 SERIALN LFDCBUF[2]                                            *
; * 05 SERIALN LFDCBUF[2]                                            *
; * 06 SERIALN LFDCBUF[2]                                            *
; * 11 CMDIMAGE LFDCBUF[0]                                           *
; * 13 CMDIMAGE LFDCBUF[2]                                           *
; *------------------------------------------------------------------
; * M E T H O D :                                                    *
; * * Diagnostic errors are generated by creating calls to the DCE FSW *
; * REPORT_DIAGNOSTIC routine -- not by trying to induce the *
; * erroneous conditions in the 8051 code that would precipitate the *
; * diagnostic reporting "naturally" but indirectly. *
; * * This script creates, as hex data, the following "assembler" code: *
; *
The object code described above is uploaded to the 8051 by means of the LFDUPLOD command. The LFDGOTO command is then used to force the 8051 to JMP to the just-uploaded code segment. After generating the diagnostics the code executes a RET instruction that should send control back to the COMMAND processor so the FSW can resume normal operation.

The presence of the HST codes in the LFDERR and the corresponding "parameter" values can be checked from the HK telemetry.

Arguments to level-0 script:
#0: 0x0A40 = Addr of REPORT_DIAGNOSTIC
#1: 0x0000 = Addr of test code

Note: 
- "#0" in the following represents the 0th argument to this script (i.e., the address of the REPORT_DIAGNOSTIC routine = 0x09D0)
- "#1" in the following represents the 1st argument to this script (i.e., the address of test code = 0x0000)

* Force Boot Mode
* Clear the diagnostic code stack.
;* Upload the program. *
;* Execute the program. *
;******************************
;
DTG 3,"(0) Sending two PORs followed by one-second WAITs"
WTO "Sending two PORs followed by one-second WAITs"
POR
WAIT 1
POR
WAIT 1
;
DTG 3,"(1) Sending LFDDIAGC; transmitting diagnostic-generation code."
WTO "Sending LFDDIAGC; transmitting diagnostic-generation code."
LFDDIAGC
WAIT 5,HK
LOG 1,LFDDIAGS,LFDBERR,LFDRERRP,LFDCPKT,LFDDLSTM,LFDCMDRX,LFDCBUF,LFMLWR,LFMUPR,LFMRGM
XMIT 1,L1
WTO "(2) Sending LFDDUPLOD"
LFDDUPLOD PROG,L1,CRC1
WAIT 5,HK
LOG 1,LFDDIAGS,LFDBERR,LFDRERRP,LFDCPKT,LFDDLSTM,LFDCMDRX,LFDCBUF,LFMLWR,LFMUPR,LFMRGM
DTG 3,"(3) Sending LFDGOTO 0x#1." 
WTO "Sending LFDGOTO 0x#1."
LFDGOTO PROG
;******************************
;* Examine Housekeeping Data *
;******************************
WAIT 5,HK
LOG 1,LFDDIAGS,LFDBERR,LFDRERRP,LFDCPKT,LFDDLSTM,LFDCMDRX,LFDCBUF,LFMLWR,LFMUPR,LFMRGM
; FORM 1,HK0,"HK Data"
CHECK 1,($LFDRERR[0] == hex("0x01"))
CHECK 1,($LFDRERR[1] == hex("0x02"))
CHECK 1,($LFDRERR[2] == hex("0x03"))
CHECK 1,($LFDRERR[3] == hex("0x04"))
CHECK 1,($LFDRERR[4] == hex("0x05"))
CHECK 1,($LFDRERR[5] == hex("0x06"))
CHECK 1,($LFDRERR[6] == hex("0x11"))
CHECK 1,($LFDRERR[7] == hex("0x13"))
;
CHECK 1,($LFDRERR[0] == $LFDCBUF[6])
CHECK 1,($LFDRERR[1] == $LFMXPFR)
CHECK 1,($LFDRERR[2] == $LFDCBUF[6])
CHECK 1,($LFDRERR[3] == $LFDCBUF[2])
CHECK 1,($LFDRERR[4] == $LFDCBUF[2])
CHECK 1,($LFDRERR[5] == $LFDCBUF[2])
CHECK 1,($LFDRERR[6] == $LFDCBUF[0])
CHECK 1,($LFDRERR[7] == $LFDCBUF[2])
DTG 3,"Test 5_1_2_5a completed successfully."
WTO "Test 5_1_2_5a completed successfully."
Appendix C. Test Report stp5_1_2_5a.rp1

<table>
<thead>
<tr>
<th></th>
<th>55555</th>
<th>1</th>
<th>222</th>
<th>55555</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssss</td>
<td>ttttt</td>
<td>pppp</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>a a</td>
<td>ssss</td>
<td>t</td>
</tr>
<tr>
<td>aaaa</td>
<td>a</td>
<td>sss</td>
<td>t</td>
<td>p</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>a a</td>
<td>sss</td>
<td>t</td>
</tr>
</tbody>
</table>

Len CRC Buffer Data
--- ---- ------ ----
002A FC69 1 74 01 12 0B 30 74 02 12 0B 30 74 03 12 0B 30 74 04 12 0B 30 74 05 12 0B 30 74 06 12 0B 30 74 07 12 0B 30 74 08 12 0B 30 74 09 12 0B 30 74 10 12 0B 30 74 11

Ver 01.13 Tue Jan 16 19:02:24 2001 "(0) Sending two PORs followed by one-second WAITs"

Ver 01.13 Tue Jan 16 19:02:26 2001 "(1) Sending LFDDIAGC; transmitting diagnostic-generation code."

LFDDIAGC

<table>
<thead>
<tr>
<th>Addr Addr HK-Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1780-179F LFDDIAGS</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>17A0-17BF</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>1640-1647 LFDFRR</td>
<td>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>1648-1657 LFDFRRP</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>1700-1701 LFCPKT</td>
<td>0000000001</td>
</tr>
<tr>
<td>1704-1704 LFDFLSTC</td>
<td>F4</td>
</tr>
<tr>
<td>170C-170D LFDFCMDX</td>
<td>0001</td>
</tr>
<tr>
<td>1718-1719 LFDFCMDR</td>
<td>0001</td>
</tr>
<tr>
<td>1664-167F LFDFCBUF</td>
<td>F4F4 0B0B 0001 FF0E 0000 FF0F 0000 FF0F 0000 FF0F 0000</td>
</tr>
<tr>
<td>177C-177D LFMLWR</td>
<td>0000</td>
</tr>
<tr>
<td>177E-177F LFMPRFR</td>
<td>0000</td>
</tr>
<tr>
<td>16FC-16FD LFMPROM</td>
<td>C001</td>
</tr>
</tbody>
</table>

LFDFUPLCD PROG,L1,CRC1

<table>
<thead>
<tr>
<th>Addr Addr HK-Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1780-179F LFDDIAGS</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>17A0-17BF</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>1640-1647 LFDFRR</td>
<td>00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>1648-1657 LFDFRRP</td>
<td>0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000</td>
</tr>
<tr>
<td>1700-1701 LFCPKT</td>
<td>0000000004</td>
</tr>
<tr>
<td>1704-1704 LFDFLSTC</td>
<td>AD</td>
</tr>
<tr>
<td>170C-170D LFDFCMDX</td>
<td>0002</td>
</tr>
<tr>
<td>1718-1719 LFDFCMDR</td>
<td>0002</td>
</tr>
</tbody>
</table>
CHECK: ($LFDERR[0] == hex("0x01"))
eval: (0000[0] == hex("0x01"))
SUCCESS

CHECK: ($LFDERR[1] == hex("0x02"))
eval: (0000[1] == hex("0x02"))
SUCCESS

CHECK: ($LFDERR[2] == hex("0x03"))
eval: (0000[2] == hex("0x03"))
SUCCESS

CHECK: ($LFDERR[3] == hex("0x04"))
eval: (0000[3] == hex("0x04"))
SUCCESS

CHECK: ($LFDERR[4] == hex("0x05"))
eval: (0000[4] == hex("0x05"))
SUCCESS

CHECK: ($LFDERR[5] == hex("0x06"))
eval: (0000[5] == hex("0x06"))
SUCCESS

CHECK: ($LFDERR[6] == hex("0x11"))
eval: (0000[6] == hex("0x11"))
SUCCESS
CHECK: ($LFDERR[7] == hex("0x13"))
eval: (0000[7] == hex("0x13"))
SUCCESS

CHECK: ($LFDERRP[0] == $LFDCBUF[6])
eval: (0000[0] == 0000[6])
SUCCESS

CHECK: ($LFDERRP[1] == $LFMXFER)
eval: (0000[1] == FC69)
SUCCESS

CHECK: ($LFDERRP[2] == $LFDCBUF[6])
eval: (0000[2] == 0000[6])
SUCCESS

CHECK: ($LFDERRP[3] == $LFDCBUF[2])
eval: (0000[3] == 0000[2])
SUCCESS

CHECK: ($LFDERRP[4] == $LFDCBUF[2])
eval: (0000[4] == 0000[2])
SUCCESS

CHECK: ($LFDERRP[5] == $LFDCBUF[2])
eval: (0000[5] == 0000[2])
SUCCESS

CHECK: ($LFDERRP[6] == $LFDCBUF[0])
eval: (0000[6] == 0000[0])
SUCCESS

CHECK: ($LFDERRP[7] == $LFDCBUF[2])
eval: (0000[7] == 0000[2])
SUCCESS

Ver 01.13 Tue Jan 16 19:02:33 2001 "Test 5_1_2_5a completed successfully."
Appendix D. Test Report stp5_1_2_5a.rp2

Ver 01.13 Tue Jan 16 19:02:24 2001  "(0) Sending two PORs followed by one-second WAITs"

--------------------- POR PACKET ---------------------
80000000

--------------------- POR PACKET ---------------------
80000000

Ver 01.13 Tue Jan 16 19:02:26 2001  "(1) Sending LFDDIAGC; transmitting diagnostic-generation code."

--------------------- COMMAND PACKET ---------------------

PARM4  PARM3  PARM2  PARM1  PARM0
045AFFFE 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFE 04480000

SN  OPCODE
0446FFFD 04440002 04427F7F 04408080

--------------------- COMMAND PACKET ---------------------

PARM4  PARM3  PARM2  PARM1  PARM0
045AFFFE 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFE 04480000

SN  OPCODE
0446FFFD 04440002 04427F7F 04408080

--------------------- COMMAND PACKET ---------------------

PARM4  PARM3  PARM2  PARM1  PARM0
045AFFFE 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFE 04480000

SN  OPCODE
0446FFFC 04440003 04427F7F 04408080
UPLOAD PACKET

00400174 00420B12 00447430 00461202 0048300B 004A0374 004C0B12 004E7430 00501204 0052300B 00540574 00560B12 00587430 005A1206 005C300B 005E1174 00600B12 00627430 00641213 0066300B 00680022

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 04523096 0450FC69 044EFFD5 044C002A 044A3FFF 0448C000
SN OPCODE 0446FFFFB 04440004 04425252 0440ADAD

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
SN OPCODE 0446FFFFA 04440005 04427F7F 04408080

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
SN OPCODE 0446FFFA 04440006 04427F7F 04408080

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
SN OPCODE 0446FFFF9 04440008 04427F7F 04408080

Ver 01.13 Tue Jan 16 19:02:31 2001 "(3) Sending LFDGOTO 0xC000."

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
SN OPCODE 0446FFFE 04440007 04421515 0440EABA

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
SN OPCODE 0446FFFF7 04440008 04427F7F 04408080

COMMAND PACKET

PARM4 PARM3 PARM2 PARM1 PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
Ver 01.13  Tue Jan 16 19:02:33 2001  "Test 5_1_2_5a completed successfully."