**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

| Letter | ECO No. | Description | Check | Approved | Date |
|---|---|---|---|---|---|
| - | | Initial Release | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

| Original Release Name | Date |
|---|---|
| Drawn: K. Brownsberger | 2-13-01 |
| Reviewed: | |
| Approved: | |
| | |
| | |

**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

| Size | Code Indent No. | Document No. | Rev |
|---|---|---|---|
| A | | COS-03-0054 | - |
| Scale: N/A | | | |

# Table of Contents

# 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**

| Error Code (Hex) | FSW Variable | HK Mnemonic |
|---|---|---|
| 01 | Upload Length | **LFDCBUF**[6] |
| 02 | Data Transfer CRC | **LFMXFER** |
| 03 | Download Length | **LFDCBUF**[6] |
| 04 | Serial Number | **LFDCBUF**[2] |
| 05 | Serial Number | **LFDCBUF**[2] |
| 06 | Serial Number | **LFDCBUF**[2] |
| 11 | Serial Number | **LFDCBUF**[2] |
| 13 | Serial Number | **LFDCBUF**[2] |
| 17 | Time | **LFCTIME** |
| 1B | Time | **LFCTIME** |
| 1C | Time | **LFCTIME** |
| 1F | Time | **LFCTIME** |
| 2F | Opcode | **LFDCBUF**[0] |
| 31 | **MBADISR** | |
| 32 | **MBADREGION** | |

## 1.6 TEST SCRIPT IMPLEMENTATION

### 1.6.1 Test Script Arguments

The script is parameterized as shown in the following Table:

**Table 1-2: Parameters/Arguments for stp5_1_2_5a.tst**

| Parameter | Meaning | Correct Argument for Version 1.13 |
|---|---|---|
| #0 | Absolute hex storage address of FSW **REPORT_DIAGNOSTIC** routine | 0B30 |
| #1 | Absolute hex storage address of 8051 "scratch" area for special code | C000 |

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:

| EGSE | | | DCE | | |
|---|---|---|---|---|---|
| **taiyo** | **shorty** | **Ginger** | **ETU** | **DCE #1** | **DCE #2** |
| X | | | | X | |

## 3.3 DATA/SOFTWARE

The following files must be present:

**Table 3-1: Required Program and Data Files**

| EGSE (shorty) Directory | File | Description |
|---|---|---|
| **\disks\galex\users\galex\tcs\uniscript** | **UniScript.pl** | **UniScript** interpreter |
| **\disks\galex\users\galex\tcs\uniscript\stp5_1_2_5a** | **u** | Shell script for this procedure |
| Ditto | **stp5_1_2_5a.tst** | Test script for this procedure (Appendix B) |

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

| Reference | Document Number | Title |
|---|---|---|
| 1 | NHB 1700.1(V1-A) | *NASA Basic Safety Manual* |
| 2 | COS-03-0054 | *DCE FSW Test Procedure 5.1.2.5a* (This document) |
| 3 | UCB-COS-008 | *COS FUV Detector Software Test Plan* |
| 4 | UCB-COS-DOC-1118 | *COS EGSE Startup Procedure* |

## 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 **UniScript**, to the Telnet terminal is represented in typeface. *Input* from the Test Conductor is represented in**-Bold** typeface.

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 1 | C:\tcs\us> **telnet taiyo.ssl.berkely.edu** | Establish connection to **taiyo** via Telnet client program |
| 2 | Login: **xxx**<br>Password: **--------** | Using telnet window, login as user **tcs** |

#### 4.1.4 Set Current Directory

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 3 | tcs@taiyo% **cd ~galex/tcs**<br>tcs@taiyo% **pwd**<br>/disks/galex/users/galex/tcs | Change current directory as shown |

4.1.5 **slogin** as **eagcos**

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

4.1.6 Set Current Directory

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 5 | eagcos:taiyo% **cd /disks/galex/users/galex/tcs/uniscript/stp5_1_2_5a**<br>eagcos:taiyo% **pwd**<br>/disks/galex/users/galex/tcs/uniscript/stp5_1_2_5a | Change current directory as shown |

4.1.7 Ensure that Proper Files are Present

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 6 | eagcos@taiyo% **ls –l**<br>Total 12<br>-rw-r--r-- 1 tcs eag 1398 Oct 8 18:03 stp5_1_2_5a.tst<br>-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

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 7 | eagcos:taiyo% **set path=($path ~dbb/scripts/bin)**<br>eagcos:taiyo% **pkill cosnoopy** | Set path as shown to enable access to **hks** tools; disable automatic generation of **LFDNOOP**s |

4.2.2 Execute the Script

| Step | Operator Entry/System Response | Description |
|---|---|---|
| 8 | eagcos:taiyo% **sh u**<br><br>$pstring=0B30,C000,0,0,0,0,0,0<br>Parameters are: Script File: stp5_1_2_5a<br>#0: 0B30<br>#1: C000<br>#2: 0<br>#3: 0<br>#4: 0<br>#5: 0<br>#6: 0<br>#7: 0<br>Report file<br><br>>/disks/galex/users/galex/tcs/ver_1_13/stp5_1_2_5a/stp5_1_2_5a.rp1<br>successfully opened.<br>Report file<br><br>>/disks/galex/users/galex/tcs/ver_1_13/stp5_1_2_5a/stp5_1_2_5a.rp2<br>successfully opened.<br>Script file<br><br>/disks/galex/users/galex/tcs/ver_1_13/stp5_1_2_5a/stp5_1_2_5a.tst<br>successfully opened at level 0.<br><br>"Address of REPORT_DIAGNOSTIC routine is 0x0B30."<br>"Address of test code is 0xC000."<br>"Are these addresses correct? If so, press Y;"<br>"If incorrect press N and edit shell for stp5_1_2_5a.tst."<br>y<br>Continuing.<br>"Sending two PORs followed by one-second WAITs"<br>"Sending LFDDIAGC; transmitting diagnostic-generation code." | Shell to **u**. You should see the accompanying output as **UniScript** executes |

| Step | Operator Entry/System Response | Description |
|---|---|---|
| | LFDDIAGC<br><br>WAIT 0: HKV0=1; HKV1=0; wc=5<br>WAIT 1: HKV1=0; wc=4<br>WAIT 1: HKV1=1; wc=3<br>"(2) Sending LFDUPLOD"<br><br>LFDUPLOD PROG,L1,CRC1<br><br>WAIT 0: HKV0=4; HKV1=2; wc=5<br>WAIT 1: HKV1=3; wc=4<br>WAIT 1: HKV1=4; wc=3<br>"Sending LFDGOTO 0xC000."<br><br>LFDGOTO PROG<br><br>WAIT 0: HKV0=7; HKV1=5; wc=5<br>WAIT 1: HKV1=6; wc=4<br>WAIT 1: HKV1=7; wc=3<br>"Test 5_1_2_5a completed successfully."<br>eagcos:taiyo% | |
| 9 | eagcos:taiyo% **cosnoopy&**<br>[1] 11065 | Re-enable generation of **LFDNOOP**s |

## 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.

SUMMARY SHEET

OPERATION TITLE:_____ WOA#_____

TEST ARTICLES IDENTIFICATION (including serial and/or part numbers):

_____

TASKS/STEPS COMPLETED: _____

_____

LOCATION: _____

TEST STARTED: TEST TERMINATED

TIME: _____ Hr/Min TIME: _____ Hr/Min

DATE:_____ DATE:_____

LOGS USED: _____

ANOMALY REPORTS GENERATED: _____

_____

COMMENTS:_____

_____

_____

_____

_____

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  **
; **  B   B  O   O  O   O    T           O   O  N   N  L      Y   Y  **
; **  B   B  O   O  O   O    T           O   O  NN  N  L       Y Y   **
; **  BBBB   O   O  O   O    T           O   O  N N N  L        Y    **
; **  B   B  O   O  O   O    T           O   O  N  NN  L        Y    **
; **  B   B  O   O  O   O    T           O   O  N   N  L        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 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 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:    *
; *                                                                      *
```

```
; * C000 7401      MOV     A,DIAG0001        ; generate               *
; * C002 1209D0    LCALL   REPORT_DIAGNOSTIC ;  DIAG0001              *
; * C005 7402      MOV     A,DIAG0002        ; generate               *
; * C007 1209D0    LCALL   REPORT_DIAGNOSTIC ;  DIAG002               *
; * C00A 7403      MOV     A,DIAG0003        ; generate               *
; * C00C 1209D0    LCALL   REPORT_DIAGNOSTIC ;  DIAG003               *
; * ...  ...       ...     ...                                        *
; * C023 7408      MOV     A,DIAG0013        ; generate               *
; * 0025 1209D0    CALL    REPORT_DIAGNOSTIC ;  DIAG013               *
; * ;                                                                *
; * 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  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: 0xC000 = Addr of test code       *
;*********************************************************************
;
ECHO     2
;
SYM      PROG    =0x#1
;
SYM      DIAG0001=0x0001
SYM      DIAG0002=0x0002
SYM      DIAG0003=0x0003
SYM      DIAG0004=0x0004
SYM      DIAG0005=0x0005
SYM      DIAG0006=0x0006
SYM      DIAG0011=0x0011
SYM      DIAG0018=0x0013
;
WTO      "Address of REPORT_DIAGNOSTIC routine is 0x#0."
WTO      "Address of test code is 0x#1."
WTO      "Are these addresses correct? If so, press Y;"
WTOR     "If incorrect press N and edit shell for stp5_1_2_5a.tst."
;
; ************************************************
; * Note: "#0" in the following represents the 0th   *
; *       argument to this script (i.e., the address *
; *       of the REPORT_DIAGNOSTIC routine = 0x09D0) *
; ************************************************
;
DATA     1, 0,0,EMPTY
DATA     1, 0,5,CONST=0x74_01_12#0
DATA     1, 5,5,CONST=0x74_02_12#0
DATA     1,10,5,CONST=0x74_03_12#0
DATA     1,15,5,CONST=0x74_04_12#0
DATA     1,20,5,CONST=0x74_05_12#0
DATA     1,25,5,CONST=0x74_06_12#0
DATA     1,30,5,CONST=0x74_11_12#0
DATA     1,35,5,CONST=0x74_13_12#0
DATA     1,40,2,CONST=0x22_00
;
LOG      1,1
;
;***********************************
;* 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,LFDERR,LFDERRP,LFCPKT,LFDLSTC,LFDCMDX,LFDCMDR,LFDCBUF,LFMLWR,LFMUPR,LFMROM
XMIT     1,L1
WTO      "(2) Sending LFDUPLOD"
LFDUPLOD PROG,L1,CRC1
WAIT     5,HK
LOG
1,LFDDIAGS,LFDERR,LFDERRP,LFCPKT,LFDLSTC,LFDCMDX,LFDCMDR,LFDCBUF,LFMLWR,LFMUPR,LFMROM
DTG      3,"(3) Sending LFDGOTO 0x#1."
WTO      "Sending LFDGOTO 0x#1."
LFDGOTO  PROG
;*****************************
;* Examine Housekeeping Data *
;*****************************
WAIT     5,HK
LOG
1,LFDDIAGS,LFDERR,LFDERRP,LFCPKT,LFDLSTC,LFDCMDX,LFDCMDR,LFDCBUF,LFMLWR,LFMUPR,LFMROM
;FORM     1,HK0,"HK Data"
CHECK    1,($LFDERR[0] == hex("0x01"))
CHECK    1,($LFDERR[1] == hex("0x02"))
CHECK    1,($LFDERR[2] == hex("0x03"))
CHECK    1,($LFDERR[3] == hex("0x04"))
CHECK    1,($LFDERR[4] == hex("0x05"))
CHECK    1,($LFDERR[5] == hex("0x06"))
CHECK    1,($LFDERR[6] == hex("0x11"))
CHECK    1,($LFDERR[7] == hex("0x13"))
;
CHECK    1,($LFDERRP[0] == $LFDCBUF[6])
CHECK    1,($LFDERRP[1] == $LFMXFER)
CHECK    1,($LFDERRP[2] == $LFDCBUF[6])
CHECK    1,($LFDERRP[3] == $LFDCBUF[2])
CHECK    1,($LFDERRP[4] == $LFDCBUF[2])
CHECK    1,($LFDERRP[5] == $LFDCBUF[2])
CHECK    1,($LFDERRP[6] == $LFDCBUF[0])
CHECK    1,($LFDERRP[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

```
                                    55555          1           222          55555
                                    5             11          2   2         5
                     ssss  ttttt  pppp   555            1              2        555
aaa
                    s        t    p   p     5           1             2            5
a   a
                    sssss    t    pppp       5          1            2              5
aaaaa
                        s    t    p      5   5          1           2           5   5
a   a
                    ssss     t    p       555   _____   111   _____  22222  _____   555
a   a

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 11
                         12 0B 30 74 13 12 0B 30 22 00

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

Addr Addr HK-Name        Value
---- ---- -------------- -----
1780-179F LFDDIAGS       0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
0000  0000  0000  0000  0000
17A0-17BF                0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
0000  0000  0000  0000  0000

1640-1647 LFDERR         00 00 00 00 00 00 00 00

1648-1657 LFDERRP        0000  0000  0000  0000  0000  0000  0000  0000

1700-1703 LFCPKT         00000001
1704-1704 LFDLSTC        F4
170C-170D LFDCMDX        0001
1718-1719 LFDCMDR        0001

1664-167F LFDCBUF        F4F4  0B0B  0001  FFFE  0000  FFFF  0000  FFFF  0000  FFFF  0000
FFFF  0000  FFFF

177C-177D LFMLWR         0000
177E-177F LFMUPR         0000
16FC-16FD LFMROM         C001

LFDUPLOD PROG,L1,CRC1

Addr Addr HK-Name        Value
---- ---- -------------- -----
1780-179F LFDDIAGS       0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
0000  0000  0000  0000  0000
17A0-17BF                0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
0000  0000  0000  0000  0000

1640-1647 LFDERR         00 00 00 00 00 00 00 00

1648-1657 LFDERRP        0000  0000  0000  0000  0000  0000  0000  0000

1700-1703 LFCPKT         00000004
1704-1704 LFDLSTC        AD
170C-170D LFDCMDX        0002
1718-1719 LFDCMDR        0002
```

```
1664-167F LFDCBUF        ADAD  5252  0004  FFFB  C000  3FFF  002A  FFD5  FC69  0396  0000
FFFF  0000  FFFF

177C-177D LFMLWR         0000
177E-177F LFMUPR         0000
16FC-16FD LFMROM         C001

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

LFDGOTO  PROG

Addr Addr HK-Name        Value
---- ---- -------------- -----
1780-179F LFDDIAGS       0913  0811  0706  0605  0504  0403  0302  0201  0000  0000  0000
0000  0000  0000  0000  0000
17A0-17BF                0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
0000  0000  0000  0000  0000

1640-1647 LFDERR         01 02 03 04 05 06 11 13

1648-1657 LFDERRP        0000  FC69  0000  0007  0007  0007  EAEA  0007

1700-1703 LFCPKT         00000007
1704-1704 LFDLSTC        EA
170C-170D LFDCMDX        0003
1718-1719 LFDCMDR        0003

1664-167F LFDCBUF        EAEA  1515  0007  FFF8  C000  3FFF  0000  FFFF  0000  FFFF  0000
FFFF  0000  FFFF

177C-177D LFMLWR         0000
177E-177F LFMUPR         0000
16FC-16FD LFMROM         C001

CHECK:   ($LFDERR[0] == hex("0x01"))
eval:    (0000[0] == hex("0x01"))

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S
```

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

S U C C E S S

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

```
                                    55555          1           222          55555
                                    5             11          2   2         5
                     ssss  ttttt  pppp   555            1              2        555
aaa
                    s        t    p   p     5           1             2             5
a   a
                    sssss    t    pppp       5          1            2               5
aaaaa
                        s    t    p      5   5          1            2            5   5
a   a
                    ssss     t    p       555   _____   111   _____  22222  _____   555
a   a

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

-------------------------------------------------------------------------
                          P O R   P A C K E T
-------------------------------------------------------------------------
80000000
-------------------------------------------------------------------------

-------------------------------------------------------------------------
                          P O R   P A C K E T
-------------------------------------------------------------------------
80000000
-------------------------------------------------------------------------

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

-----------------------------------------------------------------------------------------
                             C O M M A N D   P A C K E T
-----------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
-----------------------------------------------------------------------------------------
              SN            OPCODE
0446FFFE 04440001 04420B0B 0440F4F4
-----------------------------------------------------------------------------------------


-----------------------------------------------------------------------------------------
                             C O M M A N D   P A C K E T
-----------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
-----------------------------------------------------------------------------------------
              SN            OPCODE
0446FFFD 04440002 04427F7F 04408080
-----------------------------------------------------------------------------------------


-----------------------------------------------------------------------------------------
                             C O M M A N D   P A C K E T
-----------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
-----------------------------------------------------------------------------------------
              SN            OPCODE
0446FFFC 04440003 04427F7F 04408080
-----------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------
                    U P L O A D   P A C K E T
------------------------------------------------------------------
00400174 00420B12 00447430 00461202 0048300B 004A0374 004C0B12 004E7430
00501204 0052300B 00540574 00560B12 00587430 005A1206 005C300B 005E1174
00600B12 00627430 00641213 0066300B 00680022
------------------------------------------------------------------

------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 04520396 0450FC69 044EFFD5 044C002A 044A3FFF 0448C000
------------------------------------------------------------------------------------------
               SN            OPCODE
0446FFFB 04440004 04425252 0440ADAD
------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
------------------------------------------------------------------------------------------
               SN            OPCODE
0446FFFA 04440005 04427F7F 04408080
------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
------------------------------------------------------------------------------------------
               SN            OPCODE
0446FFF9 04440006 04427F7F 04408080
------------------------------------------------------------------------------------------


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

------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044A3FFF 0448C000
------------------------------------------------------------------------------------------
               SN            OPCODE
0446FFF8 04440007 04421515 0440EAEA
------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
------------------------------------------------------------------------------------------
               SN            OPCODE
0446FFF7 04440008 04427F7F 04408080
------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------
                              C O M M A N D   P A C K E T
------------------------------------------------------------------------------------------
            PARM4             PARM3             PARM2             PARM1             PARM0
045AFFFF 04580000 0456FFFF 04540000 0452FFFF 04500000 044EFFFF 044C0000 044AFFFF 04480000
```

```
--------------------------------------------------------------------------------
               SN            OPCODE
0446FFF6 04440009 04427F7F 04408080
--------------------------------------------------------------------------------

Ver 01.13  Tue Jan 16 19:02:33 2001    "Test 5_1_2_5a completed successfully."
```