COS DCE BOOT FSW v1.09 Component Test Results
Requirement 5.2.3.2 Echoes for Command Opcode and Parameters

Date: February 13, 2001
Document Number: COS-03-0026
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>

**Original Release**

**THE UNIVERSITY OF COLORADO**

At Boulder

The Center for Astrophysics and Space Astronomy

COS DCE BOOT FSW v1.09 Component Test Results

Requirement 5.2.3.2 Echoes for Command Opcode and Parameters

<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></td>
<td>COS-03-0026</td>
<td>-</td>
</tr>
</tbody>
</table>

Scale: N/A

**Drawn:** K. Brownsberger 2-13-01

Reviewed:

Approved:
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 .................................................................................. 3
    1.6.1 Test Script Arguments .................................................................................. 3
    1.6.2 Test Script Coding ......................................................................................... 3
2. Special Instructions ..................................................................................................... 3
  2.1 Quality Assurance ............................................................................................... 3
  2.2 Safety ................................................................................................................... 3
    2.2.1 Personal Safety ............................................................................................ 3
    2.2.2 Test Article and Equipment Safety ............................................................ 3
  2.3 Contamination ....................................................................................................... 4
3. Support Requirements ................................................................................................. 4
  3.1 Personnel ............................................................................................................. 4
  3.2 Tools, Equipment, and Materials ......................................................................... 4
  3.3 Data/Software ..................................................................................................... 5
  3.4 Required Documentation ..................................................................................... 5
4. Procedure/Task Steps ................................................................................................. 5
  4.1 Pre-Operation Activities ....................................................................................... 5
    4.1.1 Make Sure that hks Tools Are Active ........................................................... 5
    4.1.2 Make Sure that the Proper ROM Is Installed .............................................. 5
    4.1.3 Log In to the EGSE ..................................................................................... 6
    4.1.4 Set Current Directory ............................................................................... 6
    4.1.5 Slogin as eagcos ......................................................................................... 6
    4.1.6 Set Current Directory ............................................................................... 6
    4.1.7 Ensure that Proper Files are Present ......................................................... 7
  4.2 Operation Execution ............................................................................................... 7
    4.2.1 Establish Initial Test Conditions ................................................................. 7
    4.2.2 Execute the Script ....................................................................................... 7
  4.3 Post-Operation Activities ...................................................................................... 8
    4.3.1 Copy Reports to PC Files and Print Them .................................................. 8
    4.3.2 Complete The Test Procedure Form ......................................................... 8
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 5.2.3.2 — Echoes for Command Opcodes and Parameters.

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

The script synthesizes an **LFDNOOP** command whose five parameters, rather than being 0x0000, as in a normal no-op, are, respectively, 0x0010, 0x0020, 0x0030, 0x0040, 0x0050. This syntactically correct (but semantically deviant) command is uplinked to the DCE. The HK data from this command are examined: the image, **LFDCBUF**, is compared, word-by-word, with the contents of Buffer 1, into which the deviant no-op was generated.

1.6 TEST SCRIPT IMPLEMENTATION

1.6.1 Test Script Arguments

The script is not parameterized.

1.6.2 Test Script Coding

The script uses standard **UniScript** commands and directives.

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

- 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.
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):

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Director</td>
<td></td>
</tr>
<tr>
<td>Test Conductor</td>
<td></td>
</tr>
<tr>
<td>Test Technician</td>
<td></td>
</tr>
<tr>
<td>QA</td>
<td></td>
</tr>
</tbody>
</table>

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:

Table 3-1: Required Program and Data Files

<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_2_3_2\</td>
<td>u</td>
<td>Shell script for this procedure</td>
</tr>
<tr>
<td>Ditto</td>
<td>stp5_2_3_2.tst</td>
<td>Test script for this procedure (Appendix B)</td>
</tr>
</tbody>
</table>

In addition, the \texttt{hks} tools must be active. Directions for activating \texttt{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>\textit{NASA Basic Safety Manual}</td>
</tr>
<tr>
<td>2</td>
<td>COS-03-0026</td>
<td>\textit{DCE FSW Test Procedure 5.2.3.2 (this document)}</td>
</tr>
<tr>
<td>3</td>
<td>UCB-COS-008</td>
<td>\textit{COS FUV Detector Software Test Plan}</td>
</tr>
<tr>
<td>4</td>
<td>UCB-COS-DOC-1118</td>
<td>\textit{COS EGSE Startup Procedure}</td>
</tr>
</tbody>
</table>

4. PROCEDURE/TASK STEPS

4.1 PRE-OPERATION ACTIVITIES

4.1.1 Make Sure that \texttt{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

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

### 4.1.4 Set Current Directory

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td><code>tcs@shorty% cd ~galex/tcs</code>&lt;br&gt;<code>tcs@shorty% pwd /disks/galex/users/galex/tcs</code></td>
<td>Change current directory as shown</td>
</tr>
</tbody>
</table>

### 4.1.5 Slogin as eagcos

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td><code>tcs@shorty% slogin -l eagcos shorty.ssl.berkeley.edu</code>&lt;br&gt;<a href="mailto:eagcos@shorty.ssl.berkeley.edu">eagcos@shorty.ssl.berkeley.edu</a>’s password: (get from SSL personnel)</td>
<td>Slogin as <code>eagcos</code>; get password from SSL personnel</td>
</tr>
</tbody>
</table>

### 4.1.6 Set Current Directory

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>eagcos:shorty% <code>cd /disks/galex/users/galex/tcs/uniscript/stp5_2_3_2</code>&lt;br&gt;eagcos:shorty% <code>pwd /disks/galex/users/galex/tcs/uniscript/stp5_2_3_2</code></td>
<td>Change current directory as shown</td>
</tr>
</tbody>
</table>
4.1.7 Ensure that Proper Files are Present

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>eagcos@shorty% ls -l</td>
<td>List files; the .tst file and the shell script u should be present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total 12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-rw-r--r-- 1 tcs eag 1398 Oct 8 18:03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>stp5_2_3_2.tst</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-rw-r--r-- 1 tcs eag 62 Oct 9 17:44 u</td>
<td></td>
</tr>
</tbody>
</table>

4.2 OPERATION EXECUTION

4.2.1 Establish Initial Test Conditions

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td>eagcos:shorty% set path=($path ~dbb/scripts/bin)</td>
<td>Set path as shown to enable access to hks tools</td>
</tr>
</tbody>
</table>

4.2.2 Execute the Script

<table>
<thead>
<tr>
<th>Step</th>
<th>QA</th>
<th>Operator Entry/System Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td>eagcos:taiyo% more &lt; u</td>
<td>Shell to u. You should see the accompanying output as UniScript executes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#!/bin/sh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>pkill cosnoopy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>perl ../UniScript.pl stp5_2_3_2 &quot;0,0,0,0,0,0,0,0&quot; cosnoopy&amp;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>eagcos:taiyo% sh u</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$pstring=0,0,0,0,0,0,0,0</td>
<td>Parameters are: Script File: stp5_2_3_2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#0: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#1: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#2: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#3: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#4: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#5: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#6: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameters are: Script File: stp5_2_3_2</td>
<td>#7: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report file</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;/disks/galex/users/galex/uniscript/stp5_2_3_2/stp5_2_3_2.rp1</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>QA</td>
<td>Operator Entry/System Response</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>successfully opened.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report file</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>/disks/galex/users/galex/tcs/uniscript/stp5_2_3_2/stp5_2_3_2.rp2</code></td>
<td>successfully opened.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Script file</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>/disks/galex/users/galex/tcs/uniscript/stp5_2_3_2/stp5_2_3_2.tst</code></td>
<td>successfully opened at level 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Sending POR&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LFDNOOP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 0: HKV0=1; HKV1=25; wc=5</td>
<td>&quot;Transmitting synthesized LFDNOOP command with five non-zero parameters&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Waiting for HK data&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WAIT 1: HKV1=2; wc=0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Test stp 5.2.3.2 completed successfully&quot;</td>
<td></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_2_3_2.tst, stp5_2_3_2.rp1, and stp5_2_3_2.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_2_3_2 "0,0,0,0,0,0,0,0"
cosnoopy&
Appendix B. Test Script stp5_2_3_2.tst

; **********************************************************************************
; * STP 5.2.3.2 *
; * ------------------------------------------------------------------------------ *
; * Echoes for Command Opcodes and Parameters *
; * ------------------------------------------------------------------------------ *
; * Send a valid Command String with Serial Number and five parameters to the DCE. *
; * Verify that the resulting HK packet contains the complete and proper Command *
; * String echo. *
; * NOTE that since no actual DCE command requires or permits 5 parameters, we use *
; * a special command synthesized in Buffer 1, that is in fact a NOOP with para- *
; * meters 0010, 0020, 0030, 0040, 0050. The S/N is 0001. *
; **********************************************************************************

SYM DELTA=1
ECHO 2
;
DTG 3,"(0) Sending POR"
WTO "Sending POR"
POR
WAIT 1
LFDNOOP
WAIT 5,HK
;
; ******************************************************
; * Synthesize goofy NOOP *
; ******************************************************

DATA 1, 0,8,CONST=0x045AFFAF_04580050
DATA 1, 8,8,CONST=0x0456FFBF_04540040
DATA 1,16,8,CONST=0x0452FFCF_04500030
DATA 1,24,8,CONST=0x044EFFDF_044C0020
DATA 1,32,8,CONST=0x044AFFEF_04480010
DATA 1,40,8,NEXT
DATA 1,48,8,CONST=0x04427F7F_04408080
;
LOG 1,1
;
; ******************************************************
; * Transmit the command to the DCE *
; ******************************************************

DTG 3,"(1) Transmitting synthesized LFDNOOP command with five non-zero parameters"
WTO "Transmitting synthesized LFDNOOP command with five non-zero parameters"
XCMD 1
;
DTG 3,"(2) Waiting for HK data"
WTO "Waiting for HK data"
WAIT 5,HK
LOG 1,LFDCEBUF,LFDIDIAGS
;
; ******************************************************
; * Verify match *
; ******************************************************

* Verify match *

CHECK 1,($LFDCEBUF[ 0] == hex(substr($B1,108,4)))
CHECK 1,($LFDCEBUF[ 1] == hex(substr($B1,100,4)))
CHECK 1,($LFDCEBUF[ 2] == hex(substr($B1, 92,4)))
CHECK 1,($LFDCEBUF[ 3] == hex(substr($B1, 84,4)))
CHECK 1,($LFDCEBUF[ 4] == hex(substr($B1, 76,4)))
CHECK 1,($LFDCEBUF[ 5] == hex(substr($B1, 68,4)))
CHECK 1,($LFDCEBUF[ 6] == hex(substr($B1, 60,4)))
CHECK 1,($LFDCEBUF[ 7] == hex(substr($B1, 52,4)))
CHECK 1,($LFDCEBUF[ 8] == hex(substr($B1, 44,4)))
CHECK 1,($LFDCEBUF[ 9] == hex(substr($B1, 36,4)))
CHECK 1,($LFDCEBUF[10] == hex(substr($B1, 28,4)))
CHECK 1,($LFDCEBUF[11] == hex(substr($B1, 20,4)))
CHECK 1,($LFDCBUF[12] == hex(substr($B1, 12, 4)))
CHECK 1,($LFDCBUF[13] == hex(substr($B1, 4, 4)))
; DTG 3,"(3) Test stp 5.2.3.2 completed successfully"
WTO "Test stp 5.2.3.2 completed successfully"
Appendix C. Test Report stp5_2_3_2.rp1

Ver 01.09 Fri Dec 1 00:31:40 2000 "(0) Sending POR"

LFDNOOP

<table>
<thead>
<tr>
<th>Len</th>
<th>CRC</th>
<th>Buffer</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0038</td>
<td>8873</td>
<td>1</td>
<td>04 5A FF AF 04 58 00 50 04 56 FF BF 04 54 00 40 04 52 FF CF 04</td>
</tr>
<tr>
<td>50 00 30 04 4E FF DF 04 4C 00 20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04 4A FF EF 04 48 00 10 04 46 FF FD 04 44 00 02 04 42 7F 7F 04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 80 80</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ver 01.09 Fri Dec 1 00:31:41 2000 "(1) Transmitting synthesized LFDNOOP command with five non-zero parameters"

Ver 01.09 Fri Dec 1 00:31:41 2000 "(2) Waiting for HK data"

Addr | Addr | HK-Name | Value |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1664-167F</td>
<td>LFDCBUF</td>
<td>8080</td>
<td>7F7F</td>
</tr>
<tr>
<td>0000</td>
<td>0002</td>
<td>FFPD</td>
<td>0010</td>
</tr>
<tr>
<td>FFEF</td>
<td>0020</td>
<td>FFDF</td>
<td>0030</td>
</tr>
<tr>
<td>FFCF</td>
<td>0040</td>
<td>FFBF</td>
<td>0050</td>
</tr>
<tr>
<td>FFAF</td>
<td>1780-179F</td>
<td>LFDDIAGS</td>
<td>011B</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
</tbody>
</table>

CHECK: ($LFDCBUF[ 0] == hex(substr($B1,108,4)))
eval: (0000[ 0] == hex(substr(045A...,108,4)))
SUCCESS

CHECK: ($LFDCBUF[ 1] == hex(substr($B1,100,4)))
eval: (0000[ 1] == hex(substr(045A...,100,4)))
SUCCESS

CHECK: ($LFDCBUF[ 2] == hex(substr($B1, 92,4)))
eval: (0000[ 2] == hex(substr(045A..., 92,4)))
SUCCESS

CHECK: ($LFDCBUF[ 3] == hex(substr($B1, 84,4)))
eval: (0000[ 3] == hex(substr(045A..., 84,4)))
SUCCESS

CHECK: ($LFDCBUF[ 4] == hex(substr($B1, 76,4)))
eval: (0000[ 4] == hex(substr(045A..., 76,4)))
SUCCESS

CHECK: ($LFDCBUF[ 5] == hex(substr($B1, 68,4)))
eval: (0000[ 5] == hex(substr(045A..., 68,4)))
SUCCESS
SUCCESS
CHECK:  ($LFDCBUF[ 6] == hex(substr($B1, 60,4)))
eval:   (0000[ 6] == hex(substr(045A..., 60,4)))
SUCCESS
CHECK:  ($LFDCBUF[ 7] == hex(substr($B1, 52,4)))
eval:   (0000[ 7] == hex(substr(045A..., 52,4)))
SUCCESS
CHECK:  ($LFDCBUF[ 8] == hex(substr($B1, 44,4)))
eval:   (0000[ 8] == hex(substr(045A..., 44,4)))
SUCCESS
CHECK:  ($LFDCBUF[ 9] == hex(substr($B1, 36,4)))
eval:   (0000[ 9] == hex(substr(045A..., 36,4)))
SUCCESS
CHECK:  ($LFDCBUF[10] == hex(substr($B1, 28,4)))
eval:   (0000[10] == hex(substr(045A..., 28,4)))
SUCCESS
CHECK:  ($LFDCBUF[11] == hex(substr($B1, 20,4)))
eval:   (0000[11] == hex(substr(045A..., 20,4)))
SUCCESS
CHECK:  ($LFDCBUF[12] == hex(substr($B1, 12,4)))
eval:   (0000[12] == hex(substr(045A..., 12,4)))
SUCCESS
CHECK:  ($LFDCBUF[13] == hex(substr($B1,  4,4)))
eval:   (0000[13] == hex(substr(045A...,  4,4)))
SUCCESS
Ver 01.09 Fri Dec 1 00:31:42 2000   "(3) Test stp 5.2.3.2 completed successfully"
Appendix D. Test Report stp5_2_3_2,rp2

Ver 01.09 Fri Dec 1 00:31:40 2000  "(0) Sending POR"

---

POR PACKET
---

Ver 01.09 Fri Dec 1 00:31:41 2000  "(1) Transmitting synthesized LFDNOOP command with five non-zero parameters"

---

COMMAND PACKET
---

Ver 01.09 Fri Dec 1 00:31:41 2000  "(2) Waiting for HK data"

---

COMMAND PACKET
---

Ver 01.09 Fri Dec 1 00:31:42 2000  "(3) Test stp 5.2.3.2 completed successfully"