

DOCSIS ATP 1.1

PHY-16.1

JUPITER 110 ATP PHY-16.1 TEST SCRIPT DATA SHEET

CM Upstream Randomizer



A DOCSIS 1.1 ATP TEST SCRIPT DATA SHEET

Jupiter 110 PHY-16.1

DAQTron, Inc.

1007-B Mansell Road, Roswell, GA 30076
Phone 770.643.1878 • Fax 770.645.6403

Jupiter 110 – PHY-16.1

“Measures the randomizer construction and randomizer seed programmability.” – *DOCSIS 1.1 ATP*

Introduction

The upstream randomizer is based on a 15-bit PRBS shift register. The seed for the randomizer is programmable from the CMTS. An upstream data packet with a known content will be generated within a packet generator and recovered at the destination port in the packet generator. The data is compared to the expected data from the packet sent.

Test Requirements

This requires a packet generator to transmit packets, a VSA/RFSAs to capture burst and demodulate QPSK and 16QAM, and a downstream sniffer to verify the CMTS populates the UCDs with the right values.

Setup

The Jupiter 200 PHY-16.1 Test Script utilizes Jupiter hardware as shown in Figure 1 of the **Jupiter 110/200 PHY Hardware Setup Data Sheet** document.

Procedure

The test performs the following actions:

1. Load configuration from the selected test environment file (see Environment File Details section).
2. It performs initialization on the following pieces of hardware or software components:
 - a. VSA/RFSAs: Resets and sets averages to 1 (Skip step if not performing Procedure 1).
 - b. CMTS: Compiles MIB for SNMP communication
 - c. DHCP: Sets modem config file to system default
 - d. TFTP: Configures parameters for communication with TFTP logs
 - e. Cable Plant: Initializes all settings to default values in Hardware.INI
 - f. RF Interface Unit: Configures paths to route CM upstream to VSA/RFSAs
 - g. Power Outlet: Initializes communication and reboots the CM.
 - h. CM: Waits for modem to fully register with CMTS

- i. CMTS: Stores current upstream parameters that will be reset at the end of the test.
 - j. Packet Generator: Opens communication with Packet Generator then configures ports and creates 1 stream (1000 bytes long). Packet Rate is set to 10 packets per second. Background pattern set to pattern listed in environment file.
 - k. TDMA Trigger: Initializes communications with hardware used to perform TDMA triggering (Skip step if not performing Procedure 1).
 - l. Packet Sniffer: If using it initializes communication with the Sniffer.
3. Perform Procedure 1 (if selected).
- a. Read Procedure 1 specific parameters from environment file.
 - b. Read modulation profile specified in the environment file and load onto CMTS.
 - c. Save original modulation profile number to be reset at end of the test.
 - d. Move the CM to specified modulation profile.
 - e. Set the upstream channel width to 200 kHz.
 - f. Force Scrambler (docsIfCmtsModScrambler) to ON (1) for short and long IUCs (5 & 6).
 - g. Repeat the following steps for 2 different scrambler seeds and then once with the scrambler turned off.
 - i. Obtain a random number between 1 and 32767. Assign this to the docsIfCmtsModScramblerSeedSet MIB for short and long IUCs.
 - ii. If configured to capture packets with sniffer, log DS UCD's for 5 seconds and save to a file.
 - iii. Setup TDMA trigger by performing:
 - 1. Wait for modem to register and pass traffic (ping packets).
 - 2. Calculate the burst duration for 1000 byte packet using the configured modulation profile.
 - 3. Start Packet Generator.
 - 4. Read current upstream frequency and use it to train the TDMA Trigger.
 - 5. Setup trigger parameters on the VSA/RFSA.
 - 6. Measure frequency error running a routine to optimize the reference level.
 - 7. Stop Packet Generator
 - iv. Start the Packet Generator.
 - v. Demodulate an upstream burst returning the first 2048 symbols of the packet.
 - vi. Look for a repeating pattern that is exactly 4 times longer than the specified background pattern (4 symbols per byte). The acquired symbols need to be checked at all four possible symbol rotations due to phase uncertainty.
 - vii. Stop Packet Generator
 - viii. For the two steps in which the scrambler is ON perform the following:
 - 1. Configure Packet Capture on the NSI port of the packet generator.
 - 2. Send 10 packets upstream
 - 3. Look in captured packets for the background pattern that was originally configured.
 - h. Move upstream back to original modulation profile.
 - i. Delete modulation profile created for this procedure.
4. Perform Procedure 2 (if selected) by performing the following:
- a. Set Packet Length on stream already configured to 64 bytes.
 - b. Read modulation profile specified in the environment file and load onto CMTS.

- c. Save original modulation profile number to be reset at end of the test.
 - d. Move the CM to specified modulation profile.
 - e. Force Scrambler (docsIfCmtsModScrambler) to ON (1) for short and long IUCs (5 & 6).
 - f. Read the Procedure 2 Test Points from the environmental file.
 - g. For each Test Point, perform the following:
 - i. Turn OFF modem
 - ii. Set and verify the following MIBS
 1. docsIfCmtsModType
 2. docsIfCmtsModPreambleLen
 - iii. If configured to capture packets with sniffer, log DS UCD's for 5 seconds and save to a file.
 - iv. Turn ON modem
 - v. Wait for modem to register (will timeout if fails).
 - vi. Send 100 Packets Upstream
 - vii. Verify number of packets received a packet generator NSI port.
 - h. Move upstream back to original modulation profile.
 - i. Delete modulation profile created for this procedure.
5. Post Driver information for all hardware used.
 6. Return CMTS upstream to its original values.
 7. Post modem software load information.

Environment File Details

The environment file for PHY-16.1 provides a lot of flexibility for configuring not only how the script runs, but also which test points to measure. There are three main sections of the file, each detailed below. The 'General' section is used to specify global script configuration. The 'Procedure 1' section defines any specific variables need for Procedure 1. There can be any number of 'Procedure 2 Test Point' sections. These sections contain information that is specific to a particular test point. The name of each test point section is in the form: "Procedure 2 Test Point X" where X is a unique number that is greater than or equal to 1.

The Environment File list below should test the ATP as written. See italicized text below for more information on each variable.

PHY-16.1.env

[General]

Procedure 1=TRUE

If TRUE performs Procedure 1, if FALSE then skips it.

Procedure 2=TRUE

If TRUE performs Procedure 2, if FALSE then skips it.

Procedure 1 Modulation Profile = TDMA QPSK

Specifies which modulation profile to use for the Procedure 1. See operator's manual for more information regarding modulation profiles.

Procedure 2 Modulation Profile = TDMA QPSK

Specifies which modulation profile to use for the Procedure 2. See operator's manual for more information regarding modulation profiles.

Caputure with Sniffer = TRUE

If TRUE creates capture files that detail the CMTS UCD. No analysis is performed on the captured files. if FALSE, no communication is made with the Sniffer.

Background Pattern = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F //**Series of bytes to look for (hex)

The background pattern to assign the stream on the packet generator. This is value must be long enough and unique enough to allow the code to robustly find it and not confuse it with another part of the packet. The background pattern is only important for Procedure 1.

[Procedure 1]

Caputure with Packet Generator = TRUE

If TRUE the script will record packets received on the NSI packet generator port. These packets will be analyzed to verify that the original background pattern still exists.

[Procedure 2 Test Point 1]

Modulation Type=QPSK

The Modulation type to assign to IUCs 5 and 6.

ModType Step Name=R2.4

The Step Name to use when posting the results of the modulation type check.

ModPreambleLen Step Name=R2.5

The Step Name to use when posting the results of the preamble length check.

Packets step Name=R2.6

The Step Name to use when posting the results of the packet transmission check.

Preamble Length=32

The Preamble Length to assign to IUCs 5 and 6.

Log File Name=down2-5.dat

The filename to use for the DS UCD capture file.

The remaining sections match the descriptions in the section above. They are added here to show the values used to run the ATP default configuration.

[Procedure 2 Test Point 2]

Modulation Type=16-QAM

ModType Step Name=R2.7

ModPreambleLen Step Name=R2.8

Packets step Name=R2.9

Preamble Length=64

Log File Name=down2-12.dat

[Procedure 2 Test Point 3]

Modulation Type=QPSK

ModType Step Name=R2.10

ModPreambleLen Step Name=R2.11

Packets step Name=R2.12

Preamble Length=128

Log File Name=down2-19.dat

[Procedure 2 Test Point 4]

Modulation Type=16-QAM

ModType Step Name=R2.13

ModPreambleLen Step Name=R2.14

Packets step Name=R2.15

Preamble Length=256

Log File Name=down2-25.dat

Limit File Details

The limit file shown below for PHY-16.1 allows user control of test pass/fail criteria without requiring access to the actual test script. One application of this limit file would be to set up 'marginal' criteria for passing, or in other words to allow for a comfort margin in modem performance before submission for certification testing. Acronyms for data format and conditions are as specified in the **DAQTron Jupiter 110/200 Operator's Guide**.

PHY-16.1.lmt

```
docsIfCmtsModScrambler,C1%O,EQ,C2
docsIfCmtsModScramblerSeedSet,C0%T,EQ,Set
docsIfCmtsModScramblerSeedValue,C1%P,EQ,C2
ScrambledData,C0%T,EQ,Pattern Not Found
UnScrambledData,C0%T,EQ,Pattern Found
docsIfUpChannelId,C0%R,GE,1
docsIfUpChannelModulationProfile,C0%P,EQ,C1
docsIfCmtsModType,C1%O,EQ,C2
docsIfCmtsModPreambleLen,C1%P,EQ,C2
docsIfCmtsModPreambleType,C1%P,EQ,C2
```

JUPITER TEST SCRIPTS

PacketResults,C0%R,GE,80
UCDSeed,C1%P,EQ,C2
UCDPreambleLength,C1%P,EQ,C2
ErrorPosted,C0%T,EQ,No Error

Tabular Report

The Tabular report lists the **PICS**, **Measurement**, **Comment**, **Pass/Fail** and **Limit** applied to the measurement. Also, there are filter and Print buttons. The filter button allows you to view all data, only the data that passed the test or only the data that failed the test.

PICS	Measurement	Comment	Pass/Fail	Limit
	Procedure 2 - R2.3 doc\FControl\Firmware	Long Data: 08 Short Data: 08	Fail	C(108) equals C(208) AND C(208) equals C(308)
	Procedure 2 - R2.4 doc\FControl\Firmware	Long Data: 098 Short Data: 098	Fail	C(1098) equals C(2098) AND C(2098) equals C(3098)
	Procedure 2 - R2.5 doc\FControl\Firmware	Long Data: 32 Short Data: 32	Fail	C(132) equals C(232) AND C(232) equals C(332)
	Procedure 2 - R2.16 CMTS Populates UCIDs with Correct Values	Profile Length is UCID: 32	Fail	C(132) equal to C(232)
QPR1C.56 QPR1C.57 QPR1C.58	Procedure 2 - CM Transmits Data	DB.8 %	Fail	C(258.8 %) greater than or equal to 88
	Procedure 2 - R2.7 doc\FControl\Firmware	Long Data: 16-QAM Short Data: 16-QAM	Fail	C(116-QAM) equals C(216-QAM) AND C(216-QAM) equals C(316-QAM)
	Procedure 2 - R2.8 doc\FControl\Firmware	Long Data: 64 Short Data: 64	Fail	C(164) equals C(264) AND C(264) equals C(364)

Figure 2. Tabular Results for Jupiter 110 PHY-16.1.

Known Issues: ATP Deviations, Assumptions and Caveats

The following list contains all the current issues. The issues can be with the ATP, 3rd Party Hardware, or DAQTron's implementation.

- DAQTron has added one additional step in Procedure 1. The ATP calls out two iterations, each using a different scrambler seed, or capturing a packet and verifying that the pattern configured in the packet generator cannot be found. The problem is that if the demodulation is not occurring properly, the pattern will never be found which might lead to false passes. To fix this DAQTron has added an additional step that turns off the scrambler. With this step the demodulation is required to find the original pattern. This insures that the fact that the patterns weren't found in the first two iterations was not due to a bad demodulation setup.

DOCSIS Test and Test Result Tables

Results: Table R1

		Pass	Fail
R1.1	CMTS MIB docsIfCmtsModScrambler works properly	o	
R1.2	CMTS operates/accepts arbitrarily seed value	o	o
R1.3	CMTS MIB docsIfCmtsModScramblerSeed works properly	o	o
R1.4	CM randomized the data?	o	o
R1.5	CMTS de-randomized the data?	o	o
R1.6	CMTS operates/accepts arbitrarily seed value	o	o
R1.7	CMTS MIB docsIfCmtsModScramblerSeed works properly	o	o
R1.8	CM randomized the data?	o	o
R1.9	CMTS de-randomized the data?	o	o

Note: If there is any failure in the previous table then further research should be performed in order to identify whom is failing. Upstream and downstream sniffer will be required to do that.

Using the files down1-7.log and down1-15.log verified:

		Pass	Fail
R1.10	CMTS populate UCDs with the right values	o	o

Results: Table R2

		Pass	Fail
R2.1	CMTS MIB docsIfUpChannelId works properly	o	o
R2.2	CMTS MIB docsIfUpChannelModulationProfile works properly	o	o
R2.3	CMTS MIB DocsIfCmtsModScrambler works properly	o	o
R2.4	CMTS MIB DocsIfCmtsModType works properly	o	o
R2.5	CMTS MIB DocsIfCmtsModPreambleLen works properly	o	o
R2.6	CM works at QPSK and preamble length 16 symbols	o	o
R2.7	CMTS MIB DocsIfCmtsModType works properly	o	o
R2.8	CMTS MIB DocsIfCmtsModPreambleLen works properly	o	o
R2.9	CM works at 16QAM and preamble length 16 symbols	o	o
R2.10	CMTS MIB DocsIfCmtsModType works properly	o	o
R2.11	CMTS MIB DocsIfCmtsModPreambleLen works properly	o	o
R2.12	CM works at QPSK and preamble length 64 symbols	o	o
R2.13	CMTS MIB DocsIfCmtsModType works properly	o	o
R2.14	CMTS MIB DocsIfCmtsModPreambleLen works properly	o	o
R2.15	CM works at 16QAM and preamble length 64 symbols	o	o

Using the files down2-5.log, down2-12.log, down2-19.log and down2-25.log verified:

		Pass	Fail
R2.16	CMTS populates UCDs with the right values	o	o