OpenCable™ HOST-POD Interface Specification

OC-SP-HOSTPOD-IF-I13-030707

ISSUED SPECIFICATION

Notice

This document is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in the document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, or fitness for a particular purpose of this document, or any document referenced herein.

Document Status Sheet

Document Control Number:	OC-SP-HOSTPOD-IF-I13-030707
Document Title:	OpenCable™ HOST-POD Interface Specification
Revision History:	INT01 Interim Specification INT02 Interim Specification – April 10, 2000 INT03 Interim Specification – July 14, 2000 INT04 Interim Specification – January 18, 2001 INT05 Interim Specification – March 07, 2001 INT06 Interim Specification – May 15, 2001 INT07 Interim Specification – August 3, 2001 I08 Issued Specification – December 21, 2001 I09 Issued Specification – March 28, 2002 I10 Issued Specification – May 24, 2002 I11 Issued Specification – November 26, 2002 I12 Issued Specification – February 10, 2003 I13 Issued Specification – July 7, 2003
Date:	July 7, 2003
Responsible Authors:	Ralph Brown, Jeffrey Krauss, Gary Logston, Paul Zimmerman, Michael Adams

Status:	~~Work in Progress~~	~~Draft~~	Issued	~~Closed~~
Distribution Restrictions:	~~author only~~	~~OpenCable Technical team~~	~~CableLabs/ Member/ Vendor~~	Public

Key to Document Status Codes:

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Table of Contents

List of Tables

List of Figures

1 SCOPE

This section is replaced by Section 1.0 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2 OVERVIEW OF OPEN CABLE HOST-POD INTERFACE

2.1 Historical Perspective (INFORMATIVE)

This section is replaced by Section 2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2.2 Advanced Cable Services (INFORMATIVE)

This section is replaced by Section 2.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2.2.1 [Intentionally Blank]

This section intentionally left blank.

2.2.2 Interactive Program Guide (IPG)

This section is replaced by Section 2.2.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2.2.3 Video-on-Demand (VOD)

This section is replaced by Section 2.2.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2.2.4 Interactive services

This section is replaced by Section 2.2.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

2.3 References

2.3.1 References (Normative)

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[1] DOCSIS Set-top Gateway (DSG) Interface Specification, SP-DSG-I01-020228

[2] OpenCable Common Download Specification, OC-SP-CDS-IF-I05-030707

[3] SCTE 28 2003 (Formerly DVS 295) HOST-POD Interface Standard

[4] OpenCable POD Copy Protection System, OC-SP-PODCP-IF-I10-030707

[5] IETF RFC 2131 Dynamic Host Configuration Protocol, March 1997

[6] IETF RFC 2132 DHCP Options and BOOTP Vendor Extensions, March 1997

2.3.2 References (Informative)

[7] CableLabs Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, SP-BPI+ I09-020830

[8] CableLabs Data-Over-Cable Service Interface Specifications, Cable Modem Telephony Return Interface Specification, SP-CMTRI-970804

[9] CableLabs Data-Over-Cable Service Interface Specifications, Cable Modem Termination System – Network Side Interface Specification, SP-CMTS-NSII01-960702.

[10] CableLabs Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification, SP-OSSIv1.1-I06-020830

[11] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFIv1.1-I09-020830

[12] SCTE 40 2001 (formerly DVS 313) Digital Cable Network Interface Specification

[13] OpenCable Application Platform Specification 1.0, OC-SP-OCAP1.0-I07-030522

[14] OpenCable Host Device Core Functional Requirements, OC-SP-HOST-CFR-I13-030707

[15] SCTE 55-1 2002 (formerly DVS 178) Digital Broadband Delivery System: Out Of Band Transport Part 1: Mode A

[16] SCTE 55-2 2002 (formerly DVS 167) Digital Broadband Delivery System: Out Of Band Transport Part 2: Mode B

3 SYSTEM ARCHITECTURE (INFORMATIVE)

3.1 Introduction

The POD/Host Interface SHALL comply with Section 4.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

3.2 Two-way Networks

This section is replaced by Section 4.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

3.3 One-way Networks

This section is replaced by Section 4.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

3.4 Two-way Networks with DOCSIS

The POD/Host Interface SHALL comply with Section 4.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4 EXTENDED CHANNEL DATA FLOWS

4.1 Internet Protocol Flows (Informative)

This section is replaced by Section 5.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4.2 Flow Examples—QPSK Modem Case (Informative)

This section is replaced by Section 5.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4.3 Flow Examples—High Speed Host Modem Case DSG Mode (Informative)

This section is replaced by Section 5.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4.4 Summary of Extended Channel Flow Requirement (Normative)

The POD/Host Interface SHALL comply with Section 5.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4.5 System/Service Information Requirements (Normative)

The POD/Host Interface SHALL comply with Section 5.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

4.6 Emergency Alert Requirements (Normative)

The POD/Host Interface SHALL comply with Section 5.6 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5 PHYSICAL INTERFACE (NORMATIVE)

5.1 PC Card Compliance

5.1.1 POD Module Port Custom Interface (0341H)

The POD/Host Interface SHALL comply with Section 6.1.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.1.2 Power Management

The POD/Host Interface SHALL comply with Section 6.1.2 and 6.1.2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.1.3 Pin Assignment

The POD/Host Interface SHALL comply with Section 6.1.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.2 POD Module Identification

The POD/Host Interface SHALL comply with Section 6.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.3 Card Information Structure

The POD/Host Interface SHALL comply with Section 6.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.4 Host-POD OOB Interface

OpenCable requires support for OOB signaling. This signaling is provided in one of two modes: Out Of Band (OOB) Mode and DOCSIS Settop Gateway (DSG) Mode [1].

5.4.1 Out Of Band (OOB) Mode

The Host RF front-end specification provides the QPSK physical layer to support OOB (downstream and upstream) communications according to SCTE 55-2 2002 [16] (formerly DVS 167) and SCTE 55-1 2002 [15] (formerly DVS 178). The data link and media access control protocols for SCTE 55-2 2002 and SCTE 55-1 2002 are implemented in the POD Module. See Figure 1 below.

The interface data rates are:

- Forward Receiver: 1.544/3.088 Mbps and 2.048 Mbps
- Reverse Transmitter: 772/1544 Ksymbol/s and 128 Ksymbol/s
  (i.e., 1.544/3.088 Mbps and 256 Kbps)

The transmit and receive interfaces for the Host-POD OOB Interface are shown in Figure 1 below. The receiver interface comprises a serial bit stream and a clock, while the transmitter interface comprises I and Q data, a symbol clock, and a transmit-enable signal. The clock signal should be transferred from the Host to the POD, as shown in Figure 1.

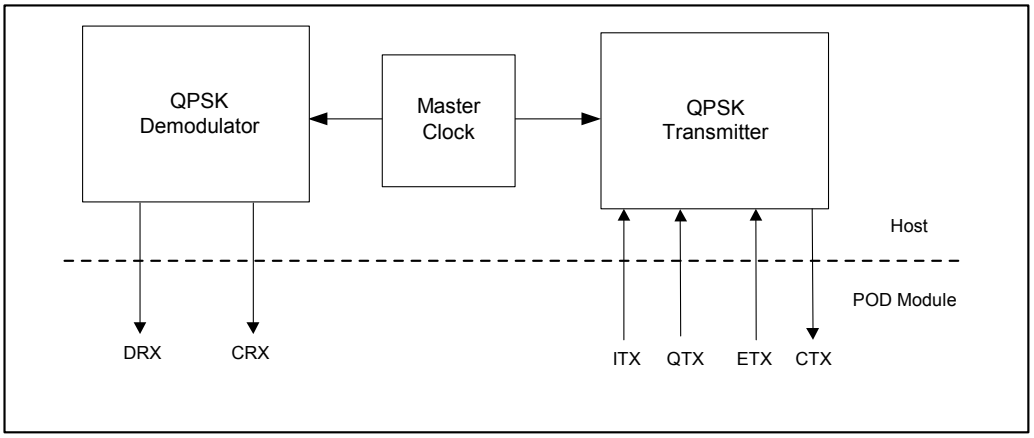


Figure 1 – Host-POD Out-of-Band Interface

The interface signals are defined below.

Table 5.4-A – Transmission Signals for Host-POD Interface

Signal	Definition	Rates	Type
DRX	RX Data	1.544/3.088 and 2.048 Mbps	I
CRX	RX Gapped Clock	1.544/3.088 and 2.048 MHz	I
ITX	TX I Channel	772/1544 and 128 Ksymbol/s	O
QTX	TX Q Channel	772/1544 and 128 Ksymbol/s	O
ETX	TX Enable	[n/a]	O
CTX	TX Gapped Symbol Clock	772/1544 and 128 KHz	I

DRX, CRX

a. DRX – The DRX data directly from the Host FDC QPSK receiver.
b. CRX – Gapped clock is a clock signal in which some of the clock cycles are missing, creating an artificial gap in the clock pattern. The clock rate is one of 1.544, 3.088 or 2.048 MHz.

ITX, QTX – Differential encoding SHALL take place within the POD module. The Host SHALL map ITX, QTX directly to the phase states shown in Figure 2 below.

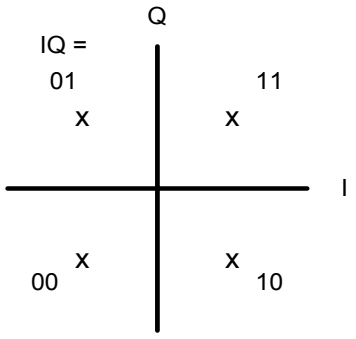


Figure 2 – Phase States for Mapping ITX and QTX

ETX – ETX is an output from the POD Module and an input to the Host. It is defined to be active high. When ETX is inactive, the values of ITX and QTX are not valid and the upstream transmitter SHALL NOT transmit such values. When ETX is active, the values of ITX and QTX are both valid and the upstream transmitter SHALL transmit these values.

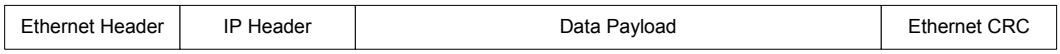
5.4.2 DOCSIS Settop Gateway (DSG) Mode

The Host DOCSIS cable modem provides the physical, data link, and media access control protocols. Unlike the OOB mode, the data link and media access control protocols for SCTE 55-1 and SCTE 55-2 are bypassed in the POD Module. The downstream communications are implemented in accordance with the DOCSIS Set-top Gateway (DSG) Specification [1]. The upstream Conditional Access Messages and network management messages will be transmitted from the POD Module via IP over the DOCSIS upstream channel using the Extended Channel.

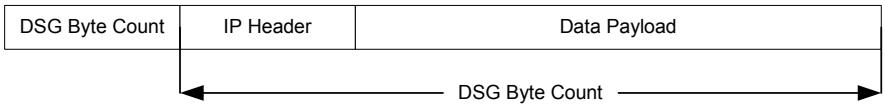
The interface data rates are:

- Downstream direction: 3.088 MBPS
- Upstream direction: Limited by DOCSIS return channel capacity

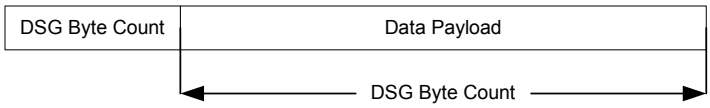
The first two bytes of the frame are the total number of bytes following in the frame, i.e., they do not include this two-byte length field. There is no CRC check required on the frame, as the interface between the Host and POD is reliable. It is the responsibility of the POD vendor to implement error detection in the DSG encapsulated data. The POD should disregard any invalid packets received from the Host. The Host must provide buffer space for a minimum of two DSG IP packets, one for transmission to the POD and one for receiving from the DOCSIS channel. Informational note: The DSG rate limits the aggregate data rate to 2.048 Mbps to avoid buffer overflow. Figure 3 below shows how the DSG packets are transported across the POD/Host interface with and without removal of the IP header bytes. The Ethernet header and CRC are removed, then optionally the IP header is removed, and a two byte field containing the byte count of the resulting data is transmitted across the POD/Host interface.



DSG Packet From Embedded DOCSIS CM



DSG Packet Across POD/Host Interface (Remove_Header_Bytes = 0)



DSG Packet Across POD/Host Interface (Remove_Header_Bytes = IP Header Size)

Figure 3 – DSG Packet Format Across POD/Host Interface

5.4.3 Timing and Voltage Parameters

The POD/Host Interface SHALL comply with Section 6.4.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.5 CPU Interface

The POD/Host Interface SHALL comply with Section 6.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.5.1 Control Register Modification

The POD/Host Interface SHALL comply with Section 6.5.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.5.2 Status Register Modification

The POD/Host Interface SHALL comply with Section 6.5.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.6 Copy Protection on the FAT Channel

The POD/Host Interface SHALL comply with the OpenCable POD Copy Protection Interface Specification [4].

5.7 Host-POD Interface Initialization

The POD/Host Interface SHALL comply with Section 6.7 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

5.8 Mechanical Design

The POD/Host Interface SHALL comply with Section 6.8 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

6 LINK INTERFACE (NORMATIVE)

6.1 Data Channel

The POD/Host Interface SHALL comply with Section 7.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

6.2 Extended Channel

The POD/Host Interface SHALL comply with Section 7.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

6.2.1 Maximum PDUs

The POD/Host Interface SHALL comply with Section 7.2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7 APPLICATION INTERFACE (NORMATIVE)

7.1 Scope Introduction

The POD/Host Interface SHALL comply with Section 8.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the exception that Table 8.1-A is replaced by Table 7.1-A below.

Table 7.1-A – Host-POD Interface Resources

Resource	EIA 679-B	OpenCable HOST-POD	Class	Type	Version	Resource ID
Resource Manager	Yes	Yes	1	1	1	00010041
MMI	Yes	Updated	64	2	1	00400081
Application Info	Yes	Updated	2	2	1	00020081
Low Speed Communication (Cable Return	Yes[2]	Updated and Optional	96	50	3	06050432
Low Speed Communication (Host Modem)	Yes[2]	Updated and Optional	96	80	3	0608043
Conditional Access Support	Yes	Yes	3	1	2	00030042
Copy Protection	No	Yes	176	3	1	00B000C1
Host Control – Info Resource	Yes	aa	32	1	3	00200043
Extended Channel Support	No	Yes	160	1	1	00A00041
Generic IPPV Support	Yes	Updated and Optional[3]	128	2	1	00800081
Specific Application Support	No	Yes	144	1	1	00900041
Generic Feature Control	No	Yes	42	1	1	002A0041
Homing Resource[1]	No	Yes	17	1	2	00110042
System Time	Yes	Yes	36	1	1	00240041
Generic Diagnostic Resource	No	Yes	260	1	1	01040041

NOTES:
1. The Homing resource is defined in SCTE-DVS/267r1, Point-of-Deployment (POD) Module Firmware Upgrade Host Interface.
2. The Resource identifier delivered by a Host SHALL be either 0x00605043 for a Host device with Cable Return Channel, or 0x00608043 for a Host with a Host Modem (e.g. DOCSIS). If no Low Speed Communication Resource Identifier reported by the Host then the Host device is assumed to be a FDC only. The POD MAY utilize the presence of this resource identifier as a means to identify what type of Cable Return Channel is supported by the Host..
3. If a device manufacturer opts to implement an optional resource on a device, then the resource must support the baseline resource ID.

7.2 Resource Manager

The POD/Host Interface SHALL comply with Section 8.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3 Man Machine Interface

7.3.1 Introduction

The POD/Host Interface SHALL comply with Section 8.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.2 Open_mmi_req() & Open_mmi_cnf()

The POD/Host Interface SHALL comply with Section 8.3.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.2.1 Open_mmi_req()

The POD/Host Interface SHALL comply with Section 8.3.2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.2.2 Open_mmi_cnf()

The POD/Host Interface SHALL comply with Section 8.3.2.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.3 Close_mmi_req() & Close_mmi_cnf()

The POD/Host Interface SHALL comply with Section 8.3.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.3.1 Close_mmi_req()

The POD/Host Interface SHALL comply with Section 8.3.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.3.3.2 Close_mmi_cnf()

The POD/Host Interface SHALL comply with Section 8.3.3.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4 Application Information

7.4.1 Introduction

The POD/Host Interface SHALL comply with Section 8.4.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4.2 Application_info_req() & Application_info_cnf()

The POD/Host Interface SHALL comply with Section 8.4.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4.2.1 Application_info_req()

The POD/Host Interface SHALL comply with Section 8.4.2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4.2.2 Application_info_cnf ()

The POD/Host Interface SHALL comply with Section 8.4.2.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3] with the addition of revised application_type values originally defined in Table 8.4-N of SCTE 28. Specifically, application_type = 0x01 has been redefined to be the "POD-Host binding information application" and application_type = 0x05 has been redefined to be the "Copy Protection application".

Application_type
The POD module application types are:

Table 7.4-N – Application Type Values

Application_type	Value (hex)	Description
Conditional Access	00	conditional access application
Host / POD Pairing Information	01	POD-Host binding information application (as defined in OC-SP-PODCP-IF)
IP Service	02	support for bi-directional IP transactions over the extended channel
Network Interface - DVS/167	03	support for DVS/167 PHY and MAC layers on the out-of-band channel
Network Interface - DVS/178	04	support for DVS/178 PHY and MAC layers on the out-of-band channel
Copy Protection	05	copy protection application
Diagnostic	06	Diagnostic application
Undesignated	07	Undesignated application
Reserved	08-FF	

7.4.3 Server_Query() & Server_Reply()

The POD/Host Interface SHALL comply with Section 8.4.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4.3.1 Server Query

The POD/Host Interface SHALL comply with Section 8.4.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.4.3.2 Server Reply

The POD/Host Interface SHALL comply with Section 8.4.3.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.5 Low Speed Communication

The Low Speed Communication resource is used to support the identification of the Forward Data Channel (FDC), the Reverse Data Channel (RDC), and any type of Host modem implementations. The Low Speed Communication resource is not a means for passing upstream/downstream OOB data to/from POD via the Host-POD interface. All downstream OOB data shall be passed directly to/from the POD via the Host-POD OOB Interface. Support for Section 8.7 of EIA-679B Part B is not permitted.

Table 7.5-A – Low Speed Communication Resource

Resource	Class	Type	Version	Identifier (hex)
Low_Speed_Communication (Cable Return)	96	50	3	00605043
Low_Speed_Communication (Host Modem)	96	80	3	00608043

The Low_Speed_Communication Identifier can be either 0x00605043 for a Host device with Cable Return Channel, or 0x00608043 for a Host with a Host Modem (e.g. DOCSIS). If no Low Speed Communication Resource Identifier reported by the Host then the Host device is assumed to be a FDC only.

7.6 Conditional Access

The POD/Host Interface SHALL comply with Section 8.6 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.6.1 CA_update()

The POD/Host Interface SHALL comply with Section 8.6.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.7 Copyright Protection

The POD/Host Interface SHALL comply with the OpenCable POD Copy Protection Interface Specification [4].

7.8 Host Control

The POD/Host Interface SHALL comply with Section 8.8 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.8.1 OOB_TX_tune_req() & OOB_TX_tune_cnf()

The POD/Host Interface SHALL comply with Section 8.8.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the exception that Table 8.8-D and text of SCTE 28 is replaced by Table 7.8-D and text below, the Status_field definition of SCTE 28 is replaced with text below, and Table 8.8-H of SCTE 28 is replaced by Table 7.8-H below.

Table 7.8-A – RF TX Frequency Value

Bit	7	6	5	4	3	2	1	0
	Frequency (MS)							
	Frequency (LS)							

RF_TX_frequency_value – This field defines the frequency of the RF Transmitter, in kHz.

The value 0x00 SHALL correspond to an output level of 0 dBmV.

Status_field – This field returns the status of the ***OOB_TX_tune_req().*** If the request was granted and the RF Transmitter set to the desired configuration, ***Status_field*** will be set to 0x00. If the Host is a unidirectional Host, ***Status_field*** SHALL be set to 0x01; the POD SHALL NOT attempt to perform RF transmit operations after receiving an ***OOB_TX_tune_cnf()*** with ***Status_field*** set to 0x01. If any of the parameters passed to the Host are outside of the Host Requirements specification [14], then the Host SHALL transmit the ***OOB_TX_tune_cnf()*** with ***Status_field*** set to 0x03. Otherwise ***Status_field*** will be set to one of the following values:

Table 7.8-H – Status Field Values for OOB TX Tune Confirm

Status_field	Value (hex)
Tuning granted	00
Tuning Denied – RF Transmitter not physically available	01
Tuning Denied – RF Transmitter busy	02
Tuning Denied – Invalid Parameters	03
Tuning Denied – Other reasons	04
Reserved	05-FF

7.8.2 OOB_RX_tune_req() & OOB_RX_tune_cnf()

The POD/Host Interface SHALL comply with Section 8.8.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.8.3 Inband_tune_req() & Inband_tune_cnf()

The POD/Host Interface SHALL comply with Section 8.8.3 and 8.8.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.9 Extended Channel Support

The POD/Host Interface SHALL comply with Section 8.9 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.9.1 New_flow_req() & New_flow_cnf()

The POD/Host Interface SHALL comply with Section 8.9.1of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the exception that Table 8.9-E of SCTE 28 is replaced by Table 7.9-E below, plus the addition of Table 7.9-Q.

Table 7.9-E New Flow Confirm Object Syntax

Syntax	# of bits	Mnemonic
New_flow_cnf() {		
New_flow_cnf_tag	24	Uimsbf
length_field()		
Status_field	8	Uimsbf
flows_remaining	8	Uimsbf
if (Status_field == 0) {		
FLOW_ID	24	Uimsbf
Service_type	8	Uimsbf
if (Service_type == IP_U) {		
IP_address	32	Uimsbf
flow_type	8	Uimsbf
flags	3	Uimsbf
max_pdu_size	13	Uimsbf
option_field_length	8	Uimsbf
For (i=0; i<option_field_length; i++) {		
option_byte	8	Uimsbf
}		
}		
}		
}		

- **Status_field** – This field returns the status of the New_flow_req(). If the request was granted and a new flow created, the Status_field will be set to 0x00. Otherwise it will be set to one of the following values:

  - **flow_type -** an 8-bit unsigned integer number that represents the protocol(s) supported by the POD to establish the IP-U flow. The field has the following values:

    0x00 UDP and TCP supported
    0x01 UDP only supported
    0x02-FF Reserved

  - **flags –** a 3-bit field that contains information, as defined in Table 7.9-Q, pertaining to limitations associated with the interactive network.

Table 7.9-Q Flag field definitions

BITS		
2	1	0
reserved		no_frag

- **no_frag** – a 1-bit boolean that designates if the network supports fragmantation. A value of 0_2 indicates that fragmentation is supported. A value of 1_2 indicated that fragmentation is not supported.

- **max_pdu_size** – a 13-bit unsigned integer number that designates the maximum PDU length that may be transmitted across the interface.

- **option_field_length -** an 8-bit unsigned integer number that represents the number of bytes of option field data to follow.

- **option_byte -** these bytes correspond to the options requested in the new_flow_req() message. The format of the field is as defined in RFC 2132 [6]. The end option (code 255) shall not be used.

7.9.2 Delete_flow_req() & Delete_flow_cnf()

The POD/Host Interface SHALL comply with Section 8.9.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.9.3 Lost_flow_ind() & Lost_flow_cnf()

The POD/Host Interface SHALL comply with Section 8.9.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.9.4 Inquire_DSG_mode(), Set_DSG_mode(), & DSG_Packet_Error()

The POD/Host Interface SHALL comply with Section 8.9.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.10 Generic IPPV Support

The POD/Host Interface support of the Generic IPPV resource is deprecated. If supported, it SHALL comply with Section 8.10 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.10.1 Program_req() & Program_cnf()

The POD/Host Interface SHALL comply with Section 8.10.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.10.2 Purchase_req() & Purchase_cnf()

The POD/Host Interface SHALL comply with Section 8.10.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.10.3 Cancel_req() & Cancel_cnf()

The POD/Host Interface SHALL comply with Section 8.10.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.10.4 History_req() & History_cnf()

The POD/Host Interface SHALL comply with Section 8.10.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11 Specific Application Support

The POD/Host Interface SHALL comply with Section 8.11 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the following exceptions of sections 8.11.3.1.2, 8.11.3.2 and 8.11.3.3 which are replaced by sections 7.11.3.1.2, 7.11.3.1, 7.11.3.2 and 7.11.3.3 below.

7.11.1 Specific Application Support Connectivity

The POD/Host Interface SHALL comply with Section 8.11.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11.2 Resource Identifier

The POD/Host Interface SHALL comply with Section 8.11.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11.3 Application Objects

The POD/Host Interface SHALL comply with Section 8.11.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11.3.1 Sas_Connect_Rqst() & Sas_Connect_Cnf()

The POD/Host Interface SHALL comply with Section 8.11.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11.3.1.1 Sas_Connect Rqst()

The POD/Host Interface SHALL comply with Section 8.11.3.1.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.11.3.1.2 Sas_Connect_Cnf()

Table 7.11-D – Sas_Connect_Cnf Object Syntax

Syntax	# of bits	Mnemonic
Sas_connect_cnf(){		
Sas_connect_cnf_tag	24	uimsbf
Length_field()		
Private_Host_Application_ID	64	uimsbf
Sas_session_status	8	uimsbf
}		

where:

Private_Host_Application_ID

This is a unique identifier of the private Host application.

Informative Note: There is no need to register Private_Host_Application_IDs used by different manufacturers. Applications that make use of this resource are downloaded into the Host by the cable

operator, and thus the application has knowledge of valid ID values that are expected from operator-supplied POD modules.

Sas_session_status

The status of the requested connection as defined in the following table.

Table 7.11-E – Sas_session_status

Sas_session_status	Value (Hex)
Connection established	00
Connection denied – no associated vendor-specific POD application found	01
Connection denied – no more connections available	02
Reserved	03-FF

7.11.3.2 Sas_Data_Rqst(), Sas_Data_Av(), & Sas_Data_Cnf()

Once a communication path has been established between the application pair (vendor-specific POD application, private Host application) via a SAS session, each of the applications can utilize the SAS APDUs to communicate with the other. The APDUs defined in this section are bi-directional in that they can originate from either side of the Host-POD Interface. The **Sas_Data_Rqst()** APDU is used by one application to inform the other application that it is ready to process incoming data. The application which receives this APDU responds with a SAS_**Data_Av()** APDU. When an application has data to send across the Host-POD Interface, a SAS_**Data_Av()** APDU is sent. The receiving application responds with a SAS_**Data_Cnf()** APDU to acknowledge that it is preparing to receive the available data.

7.11.3.2.1 Sas_Data_Rqst()

Table 7.11-F – Sas_Data_Rqst Object Syntax

Syntax	# of bits	Mnemonic
sas_data_rqst(){		
sas_data_rqst_tag	24	uimsbf
length_field()		
}		

7.11.3.2.2 Sas_Data_Av()

Table 7.11-G – Sas_Data_Av Object Syntax

Syntax	# of bits	Mnemonic
sas_data_av(){		
sas_data_av_tag	24	uimsbf
length_field()		
sas_data_status	8	uimsbf
transaction_nb	8	uimsbf
}		

7.11.3.2.3 Sas_Data_Av_Cnf()

Table 7.11-H – Sas_Data_Cnf Object Syntax

Syntax	# of bits	Mnemonic
sas_data_av_cnf(){		
sas_data_av_cnf_tag	24	Uimsbf
length_field()		
transaction_nb	8	uimsbf
}		

where:

Sas_data_status

The status of the available data defined in the following table.

Table 7.11-I – Sas_data_status

Sas_data_status	Value (Hex)
Data Available	00
Data Not Available	01
Reserved	02-FF

Transaction _nb

The Transaction number is issued from an 8-bit cyclic counter (1 – 255) and is used to identify each data transaction and to gain access to the available data. When data is not available, the transaction_nb will be set to zero.

7.11.3.3 Sas_Server_Query() & Sas_Server_Reply()

When data availability has been confirmed, a SAS_***Server_Query()*** APDU is sent to initiate the transfer of Application Specific data. The ***Sas_Server_Reply()*** APDU SHALL be used to respond to the query and transfer data.

7.11.3.3.1 Sas_Server_Query()

Table 7.11-J – Sas_Server_Query Object Syntax

Syntax	# of bits	Mnemonic
sas_server_query(){		
sas_server_query_tag	24	Uimsbf
length_field()		
transaction_nb	8	Uimsbf
}		

7.11.3.3.2 Sas_Server_Reply()

Table 7.11-K – Sas_Server_Reply Object Syntax

Syntax	# of bits	Mnemonic
sas_server_reply(){		
sas_server_reply_tag	24	uimsbf
length_field()		
transaction_nb	8	uimsbf
message_length	16	uimsbf
for (i =0; i< message_length; i++)		
{		
message_byte	8	uimsbf
}		
}		

7.12 Generic Feature Control Support

The POD/Host Interface SHALL comply with Section 8.12 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.1 Parameter Storage

7.12.1.1 Host

The POD/Host Interface SHALL comply with Section 8.12.1.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.1.2 POD

The POD/Host Interface SHALL comply with Section 8.12.1.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.2 Parameter Operation

7.12.2.1 Feature List Exchange

The POD/Host Interface SHALL comply with Section 8.12.2.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.3 Host to POD Module Transfer

The POD/Host Interface SHALL comply with Section 8.12.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.3.1 Headend to Host

The POD/Host Interface SHALL comply with Section 8.12.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.4 Resource Identifier

The POD/Host Interface SHALL comply with Section 8.12.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.5 Feature ID

The POD/Host Interface SHALL comply with Section 8.12.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6 Application Objects

The POD/Host Interface SHALL comply with Section 8.12.6 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.1 Feature List Request

The POD/Host Interface SHALL comply with Section 8.12.6.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.2 Feature List

The POD/Host Interface SHALL comply with Section 8.12.6.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.3 Feature List Confirmation

The POD/Host Interface SHALL comply with Section 8.12.6.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.4 Feature List Changed

The POD/Host Interface SHALL comply with Section 8.12.6.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.5 Feature Parameters Request

The POD/Host Interface SHALL comply with Section 8.12.6.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.6 Feature Parameters

The POD/Host Interface SHALL comply with Section 8.12.6.6 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.12.6.7 Feature Parameters Confirmation

When the POD module or Host receives the feature_parameter APDU, it SHALL respond with the feature parameters confirmation APDU.

Table 7.12-J – Feature Parameters Confirm Object Syntax

Syntax	# of bits	Mnemonic
Feature_parameters_cnf() {		
feature_parameters_cnf_tag	24	uimsbf
length_field()		
number_of_features	8	uimsbf
for(i=0; i<number_of_features; i++){		
feature_id	8	uimsbf
status	8	uimsbf
}		
}		

feature_parameters_tag Value = 0x9F9808

number_of_features Number of features to report

feature_ID Assigned feature ID number as defined in section 8.12.5 of SCTE 28.

status Status of feature parameter
00 Accepted
01 Denied – feature not supported
02 Denied – invalid parameter
03 Denied – other reason
04-FF Reserved

7.12.7 Feature Parameter Definition

The POD/Host Interface SHALL comply with Section 8.12.7 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.13 POD Module Firmware Upgrade

The POD/Host Interface SHALL comply with Section 8.13 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14 Generic Diagnostic Support

The POD/Host Interface SHALL comply with Section 8.14 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.1 Diagnostic_req()

The POD/Host Interface SHALL comply with Section 8.14.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.2 Diagnostic_cnf()

The POD/Host Interface SHALL comply with Section 8.14.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3 Diagnostic Report Definition

The POD/Host Interface SHALL comply with Section 8.14.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3] with the exception of section 8.14.3.2 which is replaced by section 7.14.3.2 below.

7.14.3.1 Memory Report

The POD/Host Interface SHALL comply with Section 8.14.3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3.2 Software Version Report

Software version reports SHALL contain the software version parameters associated with the Host.

Table 7.14-I – Software Version Report Syntax

Syntax	# of bits	Mnemonic
software_ver_report() {		
number_of_applications	8	uimsbf
for (i=0; i<number_of_applications; i++) {		
application_version_number	16	uimsbf
application_status_flag	8	uimsbf
application_name_length	8	uimsbf
for (j=0; j<application_name_length;j++){		
application_name_byte	8	uimsbf
}		
application_sign_length	8	uimsbf
for (j=0; j<application_sign_length; j++){		
application_sign_byte	8	uimsbf
}		
}		
}		

number_of_applications Total number of applications contained within the report.

application_version_number 16-bit version number of the application.

application_status_flag Status of the software, either active, inactive or downloading.

Table 7.14-J – Software Status Flag Values

Bit Value (Hex)	Software Status Flag
00	Active
01	Inactive
02	Downloading
03-FF	Reserved for future use.

application_name_length Designates the number of characters required to define the applications name.

application_name_byte ASCII character, 8-bits per character, of string that identifies the application.

application_sign_length Designates the number of characters required to define the application signature.

application_sign_byte ASCII character, 8-bits per character, of string that identifies the application signature.

7.14.3.3 Firmware Version Report

Firmware version reports SHALL contain the firmware version parameters associated with the Host.

Table 7.14-K – Firmware Version Report Syntax

Syntax	# of bits	Mnemonic
firmware_ver_report() {		
firmware_version	16	uimsbf
firmware_date {	32	uimsbf
firmware_year	16	uimsbf
firmware_month	8	uimsbf
firmware_day	8	uimsbf
}		
}		

firmware_version 16-bit version number of the firmware**.**

firmware_date 32-bit numerical representation, in the form of YYYYMMDD**,** which identifies the date of the firmware.

firmware_year 16-bit designation of the firmware's year.

firmware_month 8-bit numerical representation of the firmware's month.

firmware_day 8-bit numerical representation of the firmware's day.

7.14.3.4 MAC Address Report

The POD/Host Interface SHALL comply with Section 8.14.3.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3.5 FAT Status Report

The POD/Host Interface SHALL comply with Section 8.14.3.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3.6 FDC Status Report

In response to an FDC Status Report request the Host SHALL reply with an FDC Status Report, unless an error has occurred.

Table 7.14-O – FDC Status Report Syntax

Syntax	# of bits	Mnemonic
FDC_report() {		
FDC_center_frq	16	uimsbf
reserved	6	bslbf
carrier_lock_status	1	bslbf
packet_sync_status	1	bslbf
}		

FDC_center_frq Indicates the frequency of the FDC center frequency, in MHz. (Frequency = value * 0.05 + 50 MHz)

Table 7.14-P – FDC Center Frequency Value

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Frequency (MS)								Frequency (LS)							

reserved Reserved bits SHALL be set to 1111112.

carrier_lock_status Indicates if the current carrier is lock or not locked, 02 = not locked, 12 = locked.

packet_sync_status Indicates if the current FDC packets are in sync, 02 = not in sync, 12 = in sync.

7.14.3.7 Current Channel Report

The POD/Host Interface SHALL comply with Section 8.14.3.7 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3.8 1394 Port Report

The POD/Host Interface SHALL comply with Section 8.14.3.8 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.14.3.9 DVI Status Report

The POD/Host Interface SHALL comply with Section 8.14.3.9 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

7.15 Support for Common Download Specification

The POD/Host Interface SHALL comply with Sections 8.15 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

8 EIA-697-B (PART B) COMPLIANCE

The POD/Host Interface SHALL comply with Sections 3 and 3.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

APPENDIX A Operational Modes

This Appendix is replaced by Appendix A of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

APPENDIX B GLOSSARY

This Appendix is replaced by Appendix B of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

APPENDIX C POD HTML Baseline Requirements

This Appendix is replaced by Appendix C of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

APPENDIX D POD Interface Initialization

This appendix defines the interface initialization procedure between the POD module and the Host.

D.1 Descriptions

This Appendix D.1 has been replaced by Section 6.7.1 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.2 Card Information Structure Table

This Appendix D.2 has been replaced by Section 6.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.3 Configuration Option Register

This Appendix D.3 has been replaced by Section 6.7.2 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.4 Initialization Conditions

This Appendix D.4 has been replaced by Section 6.7.3 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.5 OOB Connection and Disconnection Behavior

This Appendix D.5 has been replaced by Section 6.7.4 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.6 Low Level Step by Step POD Personality Change Sequence

This Appendix D.6 has been replaced by Section 6.7.5 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

D.7 Initialization Overview

This Appendix D.7 has been replaced by Section 6.7.6 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the exception of section 6.7.6.7 in SCTE 28 on Copy Protection. The OpenCable POD Copy Protection Interface Specification [4] is the normative reference for this section.

D.8 Interrupt Operation

This Appendix D.8 has been replaced by Section 6.7.7 of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3].

APPENDIX E Error Handling

This Appendix is replaced by Appendix E of SCTE 28 (Formerly DVS 295) HOST-POD Interface Standard [3], with the following deviations to Table E.1-A, Error Handling.

	Error Condition	Failure Mechanism	Host Action	SCTE POD Module Action	Comments
1	POD READY signal does not go active	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.
4	Host sets data channel RS bit but POD fails to set FR bit within 5 second timeout.	POD	Minimum – Perform 1 PCMCIA rest, Report Error if not successful, Optional – Retry PCMCIA resets up to two times and then report error. Preferred –Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.
5	Host sets extended channel RS bit but POD fails to set FR bit within 5 second timeout.	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.

	Error Condition	Failure Mechanism	Host Action	SCTE POD Module Action	Comments
10	POD does not respond to Hosts open transport request within 5 seconds	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.
17	POD fails to respond to profile_inq within 5 seconds.	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful. Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.
38	POD fails to respond to ca_info_inq within 5 seconds.	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful. Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	Host reports error to user.

	Error Condition	Failure Mechanism	Host Action	SCTE POD Module Action	Comments
53	POD fails to respond to any request within 5 seconds	POD	Minimum – Perform 1 PCMCIA reset, Report Error if not successful. Optional – Retry PCMCIA resets up to two times and then report error. Preferred – Perform at least 1 PCMCIA reset. Report Error if not successful and continue to perform PCMCIA resets.	None	User may see frozen picture on scrambled channels.

APPENDIX F Revision History

OC-SP-HOSTPOD-IF-I13-030707 – contains modifications per ECNs: 03-0361, 03-0386r1, 03-0401r1, 03-0404r1, 03-0418, 03-0420, 03-0429, 03-0442

OC-SP-HOSTPOD-IF-I12-030210 – contains modifications per ECNs: 02-0341r1, 02-0349

OC-SP-HOSTPOD-IF-I11-021126 – contains modifications per ECNs: 02-0265r1, 02-0283, 02-0301, 02-0311, 02-0323

OC-SP-HOSTPOD-IF-I10-020524 – contains modifications per ECN 02-0261

OC-SP-HOSTPOD-IF-I09-020328 – contains modifications per ECN01231, ECN01232a, ECN01233, ECN01235

OC-SP-HOSTPOD-IF-I08-011221 – contains modifications per ECN01219

IS-POD-131-INT07-010803 contains modifications per ECN00188, ECN01220, ECN01221a, and ECN01222.

IS-POD-131-INT06-010515 contains modifications per ECN00161, ECN00170a, ECN00184, ECN01207, ECN01209, and ECN01213

IS-POD-131-INT05-010307 contains modifications per ECN01212

IS-POD-131-INT04-010118 contains modifications per ECN00137, ECN00150, ECN00151, ECN00157, ECN00158, ECN00159, ECN00160, ECN00163, ECN00165, ECN00167, ECN00166, ECN00169, ECN00172, ECN00173, EDCN00174, ECN00175, ECN00178, ECN00179, ECN00189, ECN00190, ECN00192, ECN00193, ECN00195, ECN00197, ECN00203, ECN01206

IS-POD-131-INT03-000714– contains modifications per ECN99003, ECN99007, ECN00051, ECN00054, ECN00060, ECN00080, ECN00090, ECN00124, ECN00133, ECN00136, ECN00138, ECN00143, and ECN00146.

IS-POD-131-INT02-000410 – contains modifications per ECN99002, ECN99004. ECN99006, ECN99026, ECN99028, ECN00031, ECN00047, ECN00049, ECN00053, ECN00078, ECN00082, ECN00083, and ECN00091.