



ANSI E1.20 - 2010
Entertainment Technology—RDM
Remote Device Management
Over DMX512 Networks

Document number: CP/2009-1017r2

© 2017 ESTA

[blank page]



ANSI E1.20 - 2010
Entertainment Technology—RDM
Remote Device Management
Over DMX512 Networks

Copyright 2017 ESTA. All rights reserved.
CP/2009-1017r2

Approved as an American National Standard by the ANSI Board of Standards
Review on 4 January 2011.

This standard was originally published when the Entertainment Services and Technology Association was operating under the name of PLASA North America. ESTA has reverted to its original name, and this document has been rebranded with the current corporate name and logo. No changes have been made to the contents of the standard.

Notice and Disclaimer

ESTA does not approve, inspect, or certify any installations, procedures, equipment or materials for compliance with codes, recommended practices or standards. Compliance with an ESTA standard or recommended practice is the sole and exclusive responsibility of the manufacturer or provider and is entirely within their control and discretion. Any markings, identification, or other claims of compliance do not constitute certification or approval of any type or nature whatsoever by ESTA.

ESTA neither guarantees nor warrants the accuracy or completeness of any information published herein and disclaim liability for any personal injury, property or other damage or injury of any nature whatsoever, whether special, indirect, consequential or compensatory, directly or indirectly resulting from the publication, use of, or reliance on this document.

In issuing and distributing this document, ESTA does not either (a) undertake to render professional or other services for or on behalf of any person or entity, or (b) undertake any duty to any person or entity with respect to this document or its contents. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstance.

Published By:

Entertainment Services and Technology Association
630 Ninth Avenue, Suite 609
New York, NY 10036
USA
Phone: 1-212-244-1505
Fax: 1-212-244-1502
Email: standards@esta.org

The ESTA Technical Standards Program

The ESTA Technical Standards Program was created to serve the ESTA membership and the entertainment industry in technical standards related matters. The goal of the Program is to take a leading role regarding technology within the entertainment industry by creating recommended practices and standards, monitoring standards issues around the world on behalf of our members, and improving communications and safety within the industry. ESTA works closely with the technical standards efforts of other organizations within our industry, including USITT and VPLT, as well as representing the interests of ESTA members to ANSI, UL, and the NFPA. The Technical Standards Program is accredited by the American National Standards Institute.

The Technical Standards Council (TSC) was established to oversee and coordinate the Technical Standards Program. Made up of individuals experienced in standards-making work from throughout our industry, the Council approves all projects undertaken and assigns them to the appropriate working group. The Technical Standards Council employs a Technical Standards Manager to coordinate the work of the Council and its working groups as well as maintain a "Standards Watch" on behalf of members. Working groups include: Camera Cranes, Control Protocols, Electrical Power, Event Safety, Floors, Fog and Smoke, Followspot Position, Photometrics, Rigging, and Stage Lifts.

ESTA encourages active participation in the Technical Standards Program. There are several ways to become involved. If you would like to become a member of an existing working group, as have over four hundred people, you must complete an application which is available from the ESTA office. Your application is subject to approval by the working group and you will be required to actively participate in the work of the group. This includes responding to letter ballots and attending meetings. Membership in ESTA is not a requirement. You can also become involved by requesting that the TSC develop a standard or a recommended practice in an area of concern to you.

The Control Protocols Working Group, which authored this Standard, consists of a cross section of entertainment industry professionals representing a diversity of interests. ESTA is committed to developing consensus-based standards and recommended practices in an open setting.

Table of Contents

Introduction	1
Overview	1
1 Normative References	3
2 Physical Layer	4
2.1 General	4
2.2 Electrical Specifications and Physical Layer Overview	4
2.2.1 ANSI E1.11 - EF1.0	4
2.3 Termination Requirements	4
2.4 Maintaining data link state between packets	5
2.4.1 Line Bias Networks	5
2.5 Command port reference circuit	6
2.5.1 Details of the reference circuit.	6
2.6 Default line state control	7
3 Timing	8
3.1 Controller Timing Requirements	8
3.1.1 Controller Packet Timing.....	8
3.1.2 Controller Packet spacing	9
3.1.3 Driver shutoff time	10
3.2 Responder Timing Requirements	10
3.2.1 Responder Packet Timings.....	10
3.2.2 Responder Packet spacing	10
3.2.3 Responder discovery response driver enable time	11
3.2.4 Driver shutoff time	11
3.2.5 Discovery Response MARK time.....	11
3.3 Data collisions	12
4 In-Line Devices	12
4.1 Overview	12
4.1.1 Transparent In-Line devices	12
4.2 Transparent In-Line Device Timing Requirements	12
4.2.1 Port Turnaround	13
4.2.2 Data Delay	13
4.2.3 Bit Distortion.....	13
4.2.4 BREAK timing	14
4.3 In-Line devices operating as part of Non-RDM system	14
4.4 In-line Devices as Responders	14
5 Device Addressing	15
5.1 General	15
5.2 UID Text Representation	15

5.3 Broadcast Message Addressing	16
5.4 Responders with multiple Responder Ports in a single Device.....	16
6 Message Structure.....	17
6.1 Byte Ordering	17
6.2 Packet Format	17
6.2.1 START Code.....	18
6.2.2 Sub START Code	18
6.2.3 Message Length	18
6.2.4 Destination UID.....	19
6.2.5 Source UID.....	19
6.2.6 Transaction Number (TN)	19
6.2.7 Port ID / Response Type.....	19
6.2.8 Message Count.....	20
6.2.9 Sub-Device Field.....	22
6.2.10 Message Data Block (MDB).....	22
6.2.11 Checksum	24
6.3 Response Type Field Values	25
6.3.1 Acknowledge (RESPONSE_TYPE_ACK)	25
6.3.2 Acknowledge Overflow (RESPONSE_TYPE_ACK_OVERFLOW)	25
6.3.3 Acknowledge Timer (RESPONSE_TYPE_ACK_TIMER).....	27
6.3.4 Negative Acknowledge (RESPONSE_TYPE_NACK_REASON).....	29
7 Discovery Method.....	30
7.1 General.....	30
7.2 Binary Search Tree	30
7.3 Discovery Process Steps	31
7.4 Discovery Mute Flag	31
7.4.1 Clearing of Mute Flag.....	31
7.5 Discovery Unique Branch Message (DISC_UNIQUE_BRANCH)	32
7.5.1 Response Unique Branch Message Decoding by Controller	34
7.5.2 Collisions.....	34
7.6 Discovery Mute/Un-Mute Messages.....	34
7.6.1 Control Field.....	34
7.6.2 Binding UID.....	35
7.6.3 Discovery Mute Message (DISC_MUTE)	35
7.6.4 Discovery Un-Mute Message (DISC_UN_MUTE)	36
7.7 Discovery Algorithm	37
7.8 Ongoing Device Discovery	39
8 Proxy Devices	40
8.1 General.....	40
8.2 Discovery.....	40
8.2.1 Represented Devices.....	40
8.2.2 Proxy Management.....	40

8.3 Response Messages	40
8.4 Proxy Management Messages	41
8.4.1 Get Proxied Device Count (PROXIED_DEVICE_COUNT)	41
8.4.2 Get Proxied Devices (PROXIED_DEVICES).....	42
9 Sub-Devices	43
9.1 General.....	43
9.2 Sub-Device Messages.....	43
9.2.1 Sub-Device Field.....	43
9.2.2 Using Sub-Devices	43
9.2.3 Required Sub-Device Messages	43
10 RDM Parameter Messages	44
10.1 Text Field Handling	44
10.2 Network Management Messages	44
10.2.1 Communication Status (COMMS_STATUS).....	44
10.3 Collection of Queued and Status Messages	46
10.3.1 Get Queued Message (QUEUED_MESSAGE).....	46
10.3.2 Get Status Messages (STATUS_MESSAGES).....	49
10.3.3 Get Status ID Description (STATUS_ID_DESCRIPTION)	52
10.3.4 Clear Status ID (CLEAR_STATUS_ID)	52
10.3.5 Get/Set Sub-Device Status Reporting Threshold (SUB_DEVICE_STATUS_REPORT_THRESHOLD)	53
10.4 RDM Information Messages	54
10.4.1 Get Supported Parameters (SUPPORTED_PARAMETERS).....	54
10.4.2 Get Parameter Description (PARAMETER_DESCRIPTION).....	56
10.5 Product Information Messages	58
10.5.1 Get Device Info (DEVICE_INFO).....	58
10.5.2 Get Product Detail ID List (PRODUCT_DETAIL_ID_LIST)	61
10.5.3 Get Device Model Description (DEVICE_MODEL_DESCRIPTION).....	62
10.5.4 Get Manufacturer Label (MANUFACTURER_LABEL)	62
10.5.5 Get/Set Device Label (DEVICE_LABEL).....	63
10.5.6 Get/Set Factory Defaults (FACTORY_DEFAULTS).....	64
10.5.7 Get Language Capabilities (LANGUAGE_CAPABILITIES).....	65
10.5.8 Get/Set Language (LANGUAGE)	66
10.5.9 Get Software Version Label (SOFTWARE_VERSION_LABEL)	67
10.5.10 Get Boot Software Version ID (BOOT_SOFTWARE_VERSION_ID).....	68
10.5.11 Get Boot Software Version Label (BOOT_SOFTWARE_VERSION_LABEL).....	69
10.6 DMX512 Setup Messages	69
10.6.1 Get/Set DMX512 Personality (DMX_PERSONALITY)	69
10.6.2 Get DMX512 Personality Description (DMX_PERSONALITY_DESCRIPTION).....	71
10.6.3 Get/Set DMX512 Starting Address (DMX_START_ADDRESS)	72
10.6.4 Get Slot Info (SLOT_INFO).....	73
10.6.5 Get Slot Description (SLOT_DESCRIPTION)	74
10.6.6 Get Default Slot Value (DEFAULT_SLOT_VALUE)	75
10.7 Sensor Parameter Messages.....	75
10.7.1 Get Sensor Definition (SENSOR_DEFINITION)	76

10.7.2 Get/Set Sensor (SENSOR_VALUE).....	78
10.7.3 Record Sensors (RECORD_SENSORS)	80
10.8 Power/Lamp Setting Parameter Messages.....	81
10.8.1 Get/Set Device Hours (DEVICE_HOURS)	81
10.8.2 Get/Set Lamp Hours (LAMP_HOURS).....	82
10.8.3 Get/Set Lamp Strikes (LAMP_STRIKES)	83
10.8.4 Get/Set Lamp State (LAMP_STATE)	84
10.8.5 Get/Set Lamp On Mode (LAMP_ON_MODE)	85
10.8.6 Get/Set Device Power Cycles (DEVICE_POWER_CYCLES).....	86
10.9 Display Setting Parameter Messages	87
10.9.1 Get/Set Display Invert (DISPLAY_INVERT).....	87
10.9.2 Get/Set Display Level (DISPLAY_LEVEL)	88
10.10 Device Configuration Parameter Messages	89
10.10.1 Get/Set Pan Invert (PAN_INVERT)	89
10.10.2 Get/Set Tilt Invert (TILT_INVERT).....	90
10.10.3 Get/Set Pan/Tilt Swap (PAN_TILT_SWAP).....	91
10.10.4 Get/Set Device Real-Time Clock (REAL_TIME_CLOCK).....	92
10.11 Device Control Parameter Messages.....	93
10.11.1 Get/Set Identify Device (IDENTIFY_DEVICE).....	93
10.11.2 Reset Device (RESET_DEVICE).....	94
10.11.3 Get/Set Power State (POWER_STATE)	95
10.11.4 Get/Set Perform Self Test (PERFORM_SELFTEST).....	96
10.11.5 Get Self Test Description (SELF_TEST_DESCRIPTION)	97
10.11.6 Capture Preset (CAPTURE_PRESET).....	98
10.11.7 Get/Set Preset Playback (PRESET_PLAYBACK).....	99
11 Operational Issues.....	101
11.1 Polling Intervals	101
12 Protocol Support Issues	101
Appendix A: Defined Parameters (Normative).....	102
Appendix B: Status Message ID's (Normative).....	118
Appendix C: Slot Info (Normative).....	120
Appendix D: Definitions (Normative)	122
Appendix E: Discovery Pseudo-Code Example (Informative)	126
E.1 Find Devices Function Call	126
E.2 Find Devices Pseudo-Code.....	126
Appendix F: Qualification tests for transmitter/receiver circuits used in RDM systems (Normative).....	129
F.1 Qualification Tests for Command port Transmitter Circuits.....	129
F.1.1 Notes on Figure F-1.....	129
F.2 Output tests for testing transmitters / line bias networks for RDM command ports	130
F.3 Line loading tests for command ports	130

F.4 Notes on the responder test circuit	132
F.5 Testing Responder Transmitters	132
Contact Information.....	133
Acknowledgments:.....	134

List of Tables

Table 2-1: Command Port Reference Circuit Values	7
Table 3-1: Controller Packet Timing	8
Table 3-2: Controller Packet Spacing Times.....	9
Table 3-3: Responder Packet Timing	10
Table 3-4: Responder Packet Spacing Times.....	11
Table 5-1: Unique ID (UID) Format.....	15
Table 6-1: Byte Ordering.....	17
Table 6-2: Packet Format	17
Table 6-3: Example of Packet showing Message Length Pointer to Checksum	18
Table 6-4: Message Data Block (MDB) Format	22
Table 6-5: Command Class Description	22
Table 6-6: Checksum Usage Example.....	24
Table 6-7: Response Type Field Allowable Values from Responder	25
Table 6-8: Key for Table 6-7	25
Table 7-1: DISC_UNIQUE_BRANCH Response Packet Encoding	33
Table 7-2: DISC_UNIQUE_BRANCH Response Packet Decoding	34
Table 7-3: Control Field	34
Table 10-1: Required Response to Status Requests.....	51
Table 10-2: Date and Time Ranges	93
Table A-1: Command Class Defines.....	102
Table A-2: Response Type Defines	102
Table A-3: RDM Categories/Parameter ID Defines	103
Table A-4: Status Type Defines	104
Table A-5: Product Category Defines	105
Table A-6: Product Detail Defines.....	107
Table A-7: Preset Playback Defines	111
Table A-8: Lamp State Defines.....	111
Table A-9: Lamp On Mode Defines	111
Table A-10: Self Test Defines	112
Table A-11: Power State Defines.....	112
Table A-12: Sensor Type Defines.....	112
Table A-13: Sensor Unit Defines	114
Table A-14: Sensor Unit Prefix Defines	115
Table A-15: Data Type Defines.....	116
Table A-16 Parameter Description Command Class Defines	116
Table A-17: Response NACK Reason Code Defines	117
Table B-1: Status Message Markers.....	118
Table B-2: Status Message ID Definitions	118
Table C-1: Slot Type.....	120
Table C-2: Slot ID Definitions.....	120

List of Figures

Figure 2-1: Command Port Reference Circuit.....	6
Figure 4-2: Bit Asymmetrical Delay.....	13
Figure 7-1: Binary Search.....	30
Figure 7-2: Device Discovery Process.....	38
Figure F-1: Command Port Transmitter Test Circuit.....	129
Figure F-2: Command Port Load Test Circuit.....	130
Figure F-3: Calibration Circuit.....	131
Figure F-4: Responder Test Circuit.....	132

Introduction

The Remote Device Management Protocol (RDM) permits intelligent bi-directional communication between devices from multiple manufacturers utilizing a modified DMX512 data link. RDM is an EF 1.0 implementation of ANSI E1.11.

RDM permits a console or other controlling device to discover and then configure, monitor, and manage intermediate and end-devices connected through a DMX512 network. RDM provides for intelligent control of devices on a DMX512 network, which has not been previously available outside of proprietary networks.

This standard specifies: the physical layer and timings, device discovery process and algorithms, message structure and communication.

Overview

This document specifies the physical layer requirements for handling half-duplex bi-directional communication and the timings associated with bi-directional communication.

This document addresses requirements for controllers, end devices, and In-Line devices such as DMX512 splitters/mergers and distribution systems to implement or support RDM communication.

An RDM system functions as a polled system, meaning that no intermediate or end-device will initiate communication. Only the device acting as the controller shall have the capability to initiate a response from any RDM device.

The RDM protocol makes use of an Alternate START Code (ASC) as defined in ANSI E1.11, to establish communication on a conventional DMX512 link. The first step in the RDM process is for a controller to identify all the devices that are connected to the data link. This is accomplished by performing a binary-tree search, or other type of search, to identify the internal Unique ID (UID) of all the connected devices.

Once a device has been discovered, the controller can request status messages, or get and set device parameters such as the DMX512 Address. All messages sent are addressed to the UID of the targeted device or through one of the Broadcast UID addresses.

Devices that primarily act as controllers (and distribution devices that do their own "discovery") shall be referred to as controllers in this document. Devices that typically receive and act on DMX512 data and/or act on RDM messages shall be referred to as responding devices or responders. Controllers send requests and receive responses on their command port. Responders receive commands and send responses on their response port. An in-line device will typically have one response port to receive commands and NULL START CODE packets from the controller, and one or more command ports of their own to forward this data to their responders.

Only one controller can be active on a given DMX512 link at any one time.

During normal operation, it is expected that the Discovery and Parameter messages will be interspersed with normal NULL START Code DMX512 packets.