

ANSI/CEA Standard

Control Networking Protocol Specification Part 5: Implementation Application Layer Guidelines

ANSI/CEA-709.5

September 2015



CEA
Consumer Electronics Association

Documents
www.CE.org
Document Sharing Hub

FOREWORD

This standard was developed by the Consumer Electronics Association under the auspices of the R7 Consumer Electronics Networking Committee.

Copyright © 2015 Consumer Electronics Association

CONTENTS

INTRODUCTION	1
TAGS	1
OVERVIEW	1
1 SCOPE	2
2 REFERENCES	2
2.1 NORMATIVE REFERENCES	2
2.1.1 Normative Reference List	3
2.1.2 Normative Reference Acquisition	3
3 TERMS AND DEFINITIONS	3
3.1 APPLICATION SET	3
3.2 BASE TYPE	3
3.3 CHANGEABLE-TYPE NETWORK VARIABLE	3
3.4 CONFIGURATION PROPERTY (CP) DATA VALUE USED TO CONFIGURE THE APPLICATION PROGRAM IN A DEVICE	4
3.5 CONFIGURATION-PROPERTY MEMBER	4
3.6 CONFIGURATION-PROPERTY MEMBER NUMBER	4
3.7 CONFIGURATION-PROPERTY TYPE INDEX	4
3.8 DEVICE	4
3.9 DEVICE CHANNEL ID	4
3.10 DEVICE CLASS	4
3.11 DEVICE INTERFACE	4
3.12 DEVICE-LOCATION FIELD	4
3.13 DEVICE SELF-DOCUMENTATION STRING DSDS	5
3.14 DEVICE SUBCLASS	5
3.15 DYNAMIC FUNCTIONAL BLOCK	5
3.16 DYNAMIC NETWORK VARIABLE	5
3.17 FORMAT	5
3.18 FUNCTIONAL BLOCK	5
3.19 FUNCTIONAL-BLOCK INDEX	5
3.20 FUNCTIONAL PROFILE FP	5
3.21 FUNCTIONAL-PROFILE KEY	6
3.22 FUNCTIONAL-PROFILE MEMBER	6
3.23 FUNCTIONAL-PROFILE MEMBER NUMBER	6
3.24 FUNCTIONAL-PROFILE NUMBER	6
3.25 FUNCTIONAL-PROFILE SELECTOR	6
3.26 FUNCTIONAL-PROFILE TEMPLATE	7
3.27 GLOBAL INDEX	7
3.28 INHERITING PROFILE	7
3.29 INTEROPERABILITY	7
3.30 CNP DEVICE	7
3.31 CNP NETWORK	7
3.32 MANUFACTURER ID MID	7
3.33 NETWORK-INTERFACE SELECTION	7
3.34 NETWORK VARIABLE NV	8
3.35 NETWORK-VARIABLE DECLARATION	8
3.36 NETWORK-VARIABLE INDEX	8
3.37 NETWORK-VARIABLE MEMBER	8
3.38 NETWORK-VARIABLE MEMBER NUMBER	8
3.39 NETWORK-VARIABLE PROGRAMMATIC NAME	8
3.40 NETWORK-VARIABLE SELECTION	8
3.41 NETWORK-VARIABLE TYPE	8
3.42 NETWORK-VARIABLE TYPE INDEX	9

3.43	UNIQUE NODE ID	9
3.44	NODE.....	9
3.45	PASSIVE CONFIGURATION TOOL PCT	9
3.46	PRIMARY FUNCTIONAL BLOCK	9
3.47	PRIMARY FUNCTIONAL PROFILE	9
3.48	PROPRIETARY DATA.....	9
3.49	SELF-DOCUMENTATION STRING SD STRING.....	9
3.50	SELF-DOCUMENTATION TEXT	10
3.51	SHARED-MEDIA CHANNEL.....	10
3.52	STANDARD CONFIGURATION-PROPERTY TYPE SCPT.....	10
3.53	STANDARD NETWORK-VARIABLE TYPE SNVT.....	10
3.54	STANDARD PROGRAM ID SPID	10
3.55	STATIC FUNCTIONAL BLOCK.....	10
3.56	STATIC NETWORK VARIABLE	10
3.57	SUBSYSTEM.....	10
3.58	SUCCESSFUL COMMISSIONING.....	11
3.59	SYSTEM	11
3.60	UNCONFIGURED DEVICE.....	11
3.61	USAGE	11
3.62	USAGE ID	11
3.63	USER DATA.....	11
3.64	WINK FUNCTION.....	11
4	DEVICE INTERFACES	11
4.1	GENERAL.....	11
4.2	UNIQUE NODE ID	12
4.3	STANDARD PROGRAM ID.....	13
4.3.1	GENERAL.....	13
4.4	GUIDELINE 4.3: A DEVICE SHALL IMPLEMENT A STANDARD PROGRAM ID AS DEFINED IN 4.3, STANDARD PROGRAM ID	13
4.4.1	<i>Format Field</i>	13
4.4.2	<i>Manufacturer Field</i>	13
4.4.3	<i>Device Class Field</i>	13
4.4.4	<i>Usage Field</i>	14
4.4.4.1	<i>General</i>	14
4.4.5	<i>Channel Type Field</i>	15
4.4.6	<i>Model Number Field</i>	15
4.5	DEVICE CHANNEL ID.....	15
4.6	DEVICE LOCATION FIELD.....	15
4.7	DEVICE SELF-DOCUMENTATION STRING (DSDS).....	16
4.8	FUNCTIONAL BLOCKS	17
4.8.1	<i>General</i>	17
4.8.2	<i>Implementing a Functional Block</i>	19
4.8.3	<i>Network Variables</i>	20
4.8.4	<i>Configuration Properties</i>	27
4.9	DEVICE AND FUNCTIONAL BLOCK VERSIONING.....	38
4.10	DEVICE INTERFACE (XIF) FILE	39
5	RESOURCE FILES	40
5.1	RESOURCE FILE DEFINITIONS	40
5.1.2	<i>Type Definitions</i>	41
5.1.3	<i>Functional Profiles</i>	44
5.1.4	<i>Language Strings</i>	47
5.1.5	<i>Formats</i>	48
5.2	IDENTIFYING APPROPRIATE RESOURCES.....	51

5.2.1 *Standard and User Resources* 51

5.2.2 *Using Standard Resources* 52

5.2.3 *Using User Resources*..... 52

6 NETWORK INSTALLATION..... 53

6.1 GENERAL..... 53

6.2 NETWORK ADDRESSING 54

6.2.1 *Network Addressing Scheme* 54

6.2.2 *Address-Table Entries* 55

6.2.3 *Network Variable Aliases*..... 55

6.2.4 *Domain-Table Entries* 56

6.2.5 *Self-Installed Devices* 57

6.2.6 *Field-Installed Devices* 57

6.3 PASSIVE CONFIGURATION TOOLS 58

6.4 SERVICE PIN..... 58

6.5 GATEWAYS TO COMMAND-BASED SYSTEMS..... 59

6.6 SHARED-MEDIA CONSIDERATIONS 60

ANNEX A - DEVICE RESOURCE FILES (INFORMATIVE)..... 61

Copyright Consumer Technology Association
Provided by IHS Markit under license with CTA
No reproduction or networking permitted without license from IHS

Introduction

This standard is prepared based upon work provided to the Consumer Electronics Association (CEA) R7 Consumer Electronics Networking Committee by LonMark International, a non-profit standards development association. This work has been modified from its original creation to update it to the latest revision. Similar documents have been used in other international standards bodies.

This standard is part of a series of standards for open data transmission in building automation, home automation and control, and in building management systems. The content of this standard covers the data communications used for management, automation/control and field functions.

The CEA-709.5 is part of a series of ANSI/CEA-709 Standards under the general title *Control Network Protocol (CNP)*, which comprises the following parts:

- Part 1: *Protocol Stack Specification*
- Part 2: *Power Line Channel Specification*
- Part 3: *Twisted-Pair Communication*
- Part 4: *Fiber-Optic Specification*
- Part 5: *Implementation*
- Part 6: *Application Elements*

At present this document exists only in English.

Tags

Descriptors: system management, open systems interconnection, interoperability, device communications, network interconnection, buildings, heating, ventilation, air conditioning, data bus, data transmission, protocols, device profiles, building automation, controls, building management

Overview

This standard specifies the Layered Implementation Guidelines (LIG) for the Control Network Protocol (CNP) Specification: ANSI/CEA-709.1-D. The CNP specification model is based on the ISO Open Systems Interconnection Reference Model. There are also important extensions to the 7-layer OSI Reference Model. Figure 1 shows the scope of this specification in reference to the CNP and companion specifications for handling various data-transport media at the lower ISO protocol layers. A dashed line is used to show that the scope of this standard is not as redundancy to the other specifications covering their respective layers but as a complement to those specifications in implementing them in an interoperable fashion.

In this standard, the guidelines for implementing a device based on CNP are specified to increase the ability for devices to interoperate—regardless of installer or manufacturer of the devices. Anything outside this boundary is covered in other parts of the standard. Similar specifications exist for CNP data-transport media.

This standard has been prepared to provide mechanisms through which various vendors of building automation, control, and of building-management systems, may exchange information in a standardised way. It defines communication and internal-documentation requirements.

This standard is contributing to the general US policy for energy savings for building automation, monitoring, and control of energy consuming systems in typical buildings.

ANSI/CEA-709 Control Network Protocol	CEA-709.6 Application Elements			
	CEA-709.5 Implementation Guideline			
	ANSI/CEA-709.1 Protocol Stack			
	ANSI/CEA-709.2 Power Line Channel	ANSI/CEA-709.3 Twisted-Pair Communication	ANSI/CEA-709.4 Fiber-Optic Channel	ANSI/CEA-852.1 Enhanced IP Tunneling Protocol

Figure 1 — Scope of the ANSI/CEA-709 suite of standards

1 Scope

This standard contains all the information necessary to facilitate the exchange of data and control information in an interoperable fashion using ANSI/CEA-709.1-D and its associated data-transport media specifications.

This standard establishes a minimal set of rules for compliance. It does not rule-out extended services to be provided, given that the rules are adhered-to within the system. It is the intention of the standard to permit extended services to coexist and defines the bounds in which those services function, including the format for internal device-documentation of those services. Services outside purvey of this specification—so long as they are adherents of the system—are permitted but will not necessarily be interoperable with any other devices and shall not be essential for the functioning of the device.

Certain aspects of this standard are defined in other documents. These documents are referenced where relevant. In the case where a referenced document conflicts with this standard, this standard will prevail.

2 References

2.1 Normative References

The following referenced standards are indispensable for the application of this standard. These standards contain provisions that, through reference in this text, constitute normative provisions of this standard. At the time of publication, the editions indicated were valid. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1.1 Normative Reference List

- ANSI/CEA-709.1-D - CONTROL NETWORK PROTOCOL SPECIFICATION
- ANSI/CEA-709.2 - CONTROL NETWORK POWER LINE (PL) CHANNEL SPECIFICATION
- ANSI/CEA-709.3 - FREE-TOPOLOGY TWISTED-PAIR CHANNEL SPECIFICATION
- ANSI/CEA-709.4 - FIBER-OPTIC CHANNEL SPECIFICATION
- ANSI/CEA-852.1-A - ENHANCED PROTOCOL FOR TUNNELLING COMPONENT NETWORK PROTOCOLS OVER INTERNET PROTOCOL CHANNELS
- CEA-709.6 – CONTROL NETWORKING PROTOCOL SPECIFICATION: PART 6: APPLICATION ELEMENTS

2.1.2 Normative Reference Acquisition

1. Consumer Electronics Association (CEA); 1919 South Eads Street, Arlington, VA 22202; Tel: +1 703 907 7500; <http://www.techstreet.com>

3 Terms and Definitions

For the purpose of this standard, the terms and definitions given in ANSI/CEA-709.1-D and the following apply.

3.1

application set

function block or function blocks to which a configuration property applies

EXAMPLE A network variable, a series or compilation of network variables, a functional block, a series or compilation of functional blocks, or the entire device.

3.2

base type

fundamental type that can be used as the basis of a network-variable type or configuration-property type.

NOTE The available base types are defined in section 5.1.2.2.

3.3

changeable-type network variable

network variable whose type can be changed during installation

NOTE See 4.8.3.3.

3.4

configuration property (CP)

data value used to configure the application program in a device

NOTE Configuration properties are used to set parameters such as maximum, minimum, default, and override values. CPs are implemented using configuration network variables or as data items within configuration files. Configuration-property data are kept in a device's non-volatile memory.

3.5

configuration-property member

part of a functional profile

NOTE See 3.22.

3.6

configuration-property member number

Part of a functional profile

NOTE See 3.23.

3.7

configuration-property type index

16-bit number that uniquely identifies a configuration-property type within the scope defined by the scope number and program-ID template of the resource file that contains the configuration-property type definition

3.8

device

logical and physical entity of the network containing an application that is designed to communicate with other logical and physical entities

3.9

device channel ID

number that optionally specifies the channel to which a device is attached

3.10

device class

two-byte field identifying the primary function of a device and part of the SPID of the device

3.11

device interface

network-visible interface to a device consisting of the unique node ID, program ID, channel ID, location field, device self-documentation string, device configuration properties, and functional blocks

3.12

device-location field

string or number that optionally specifies the location of a device

3.13

device self-documentation string

DSDS

string that specifies the structure of the contents of the self-documentation strings, the functional blocks, and optionally describes the function of a device

3.14

device subclass

two-byte field specifying the usage in the first byte and the channel type in the second byte and is part of the SPID of a device

NOTE See the usage and channel-type definitions.

3.15

dynamic functional block

functional block that is added to a device by a network tool after the device is installed

3.16

dynamic network variable

network variable that is added to a device by a network tool after the device is installed

3.17

format

<program ID> four-bit value defining the structure of the program ID as being a Standard Program Identifier (SPID) and device self-documentation string (DSDS) in the device

<resource file> string that provides formatting instructions for a network-variable or configuration-property type

3.18

functional block

portion of a device's application that performs a task by receiving configuration and operational data inputs, processing the data, and sending operational data outputs

NOTE A functional block may receive inputs from the network, from hardware attached to the device, and/or from other functional blocks on a device. A functional block may send outputs to the network, to hardware attached to the device, and/or to other functional blocks on the device. A functional block is an implementation of a functional profile. A "standard" functional block is one based on a standard functional profile template (SFPT).

3.19

functional-block index

sequentially assigned number identifying a functional-block implementation on a device

3.20

functional profile FP

template that describes common units of functional behaviour, also known as profiles, or FPs; which can be represented with a machine-readable functional-profile template (FPT)

NOTE Each functional profile consists of a profile description and a specified set of network variables and configuration properties designed to perform a single function on

a device. The network variables and configuration properties specified by the functional profile are called the functional-profile members. A functional profile specifies whether the implementation of each functional-profile member is mandatory or optional. A profile is uniquely identified by a program-ID template, scope, and functional-profile number.

3.21

functional-profile key

functional-profile number

NOTE See 3.24.

3.22

functional-profile member

network-variable or configuration-property member of a functional profile

NOTE Each functional-profile member is identified as mandatory or optional by the functional profile. Each member also includes a text description of the member for the functional profile.

3.23

functional-profile member number

two-byte number that uniquely identifies a network-variable or configuration-property member of a functional profile

NOTE This member number is used to associate a network variable or configuration property on a device with the corresponding network-variable or configuration-property member of the functional profile. Member numbers shall be in the range of 1 to 4095, and need not be contiguous. Member numbers shall be unique, with the exception that network-variable and configuration-property members may use the same number. (Therefore, network-variable members' numbers shall be unique, and configuration-property members' numbers shall be unique, but they need not be unique between network-variable members and configuration-property members.) There may be a maximum of 255 mandatory members and 255 optional members of each type (scope 0 NV, inheriting NV, scope 0 CP, and inheriting CP).

3.24

functional-profile number

two-byte number that uniquely identifies a functional profile within the scope defined by the scope number and program-ID template of the resource file that contains the functional-profile definition

NOTE Also called the functional-profile key, or the FPT key.

3.25

functional-profile selector

an ASCII vertical bar ("|") or an ASCII number sign ("#") to denote the association of a network variable or configuration property with a scope-0 profile or a profile of a higher-numbered scope, respectively, where a higher-numbered scope would denote the NV or CP was added to enhance a scope-0 profile or that it applies to a non-standard profile

NOTE If the functional profile selector is a vertical bar, the member number identifies a member of a scope-0 profile. If the functional profile selector is a number sign, the member number identifies a member of the inheriting profile. The number-sign functional profile selector is always used for members of user functional profiles, including profiles that do not use inheritance. The vertical-bar functional profile selector is always used for

members of standard functional profiles. Two different functional profile members may have the same member number as long as they use different functional profile selectors. EXAMPLE The “|1” member of a functional profile is not the same as the “#1” member of the same profile. This prevents conflicts if new members are added to a standard functional profile that has already been used as the basis for inheriting profiles.

3.26

functional-profile template

functional profile in human- and machine-readable form

NOTE See 3.20.

3.27

global index

functional-block index

NOTE See 3.19.

3.28

inheriting profile

functional profile that inherits members from a scope-0 profile

3.29

interoperability

conditions that ensure multiple devices—from the same or different manufacturers—can be integrated into a single network without requiring custom device or tool development

3.30

CNP device

hardware and software that runs an application and communicates with other devices using the ANSI/CEA-709.1 protocol

NOTE It may optionally interface with input/output hardware. A CNP device includes at least one processor and a CNP transceiver also called a CNP node, or simply a node.

3.31

CNP network

collection of intelligent devices that communicate with each other using the ANSI/CEA-709.1 protocol over one or more communications channels

3.32

manufacturer ID

MID

20-bit number that uniquely identifies the device manufacturer of a device and is part of the device’s SPID

3.33

network-interface selection

form of network-variable selection that occurs on the network interface

3.34

network variable

NV

data item that a particular device application program expects to get from other devices on a network (an input network variable) or expects to make available to other devices on a network (an output network variable)

NOTE Network variable data are typically stored in a device's volatile memory.

EXAMPLE Examples are a temperature, switch value, and actuator-position setting.

3.35

network-variable declaration

establishment of an instance of a network variable type within the code of an application

3.36

network-variable index

sequentially assigned number identifying a network variable implementation on a device

NOTE For Neuron C applications, the index is assigned by the Neuron C compiler in the order of declaration. The first network variable on a device has index 0, the second index 1, etc.

3.37

network-variable member

functional-profile member that is a network variable

NOTE See 3.22.

3.38

network-variable member number

number of a functional-profile member that is a network variable

NOTE See 3.23.

3.39

network-variable programmatic name

name assigned to a network-variable implementation by the device application developer

NOTE The programmatic name is limited to 16 characters, including any optional prefixes. The programmatic name is not significant for interoperability, but conventions are suggested in 4.8.3.4 to make programmatic names easier to use for integrators.

3.40

network-variable selection

process of associating a network-variable selector with a network variable on a device

3.41

network-variable type

specification of the length, units, valid range, and resolution of the data contained within a network variable

NOTE A network variable type may be a simple, one, two, or four-byte scalar type; or a more complex structure or union of up to 31 bytes.

3.42

network-variable type index

16-bit number that uniquely identifies a network-variable type within the scope defined by the scope number and program-ID template of the resource file that contains the network-variable type definition

3.43

unique node ID

unique 48-bit identifier within the read-only data structure of a device as defined by the ANSI/CEA-709.1 protocol

NOTE It is also called the `unique_node_ID`.

3.44

node

<common> device

<precise> physical and logical presence on a CNP network with a unique node ID and network address

NOTE The unique node ID relates to the identification of a single instance of an implemented ANSI/CEA-709.1 protocol stack. A device is also a network presence with an application processor and one or more nodes. A device with multiple unique node IDs would consist of multiple nodes. Some infrastructure devices, such as routers, also consist of more than one unique node ID and thus consist of multiple nodes.

3.45

passive configuration tool

PCT

network tool that can be used on a device to assist in the successful commissioning of the device without disrupting the operation of other network tools

NOTE It may be a plug-in, standalone software, hardware attachment, or other tool. A passive configuration tool has attributes and capabilities as defined in clause 6.3.

3.46

primary functional block

functional block on a device that implements the most important function for the device

3.47

primary functional profile

functional profile that defines the primary functional block on a device

3.48

proprietary data

data and message definitions in the device interface that are known only to the manufacturer and the manufacturer's agents

3.49

self-documentation string

SD string

text string associated with a device, network variable, or configuration property that is stored within a device and within the device interface (XIF) file for a device

NOTE Network tools can read the self-documentation strings from the device itself or from the device interface file.

3.50

self-documentation text

optional text within a device, network variable, or configuration property self-documentation string that provides documentation of the intended use of the device, network variable, or configuration property respectively for use by integrators

3.51

shared-media channel

communications channel where messages can leak between tools and devices belonging to different systems

3.52

standard configuration-property type

SCPT

configuration-property type that has been standardized by the CEA-709.6

NOTE A SCPT is a standardized definition of the units, scaling, encoding, valid range, and meaning of the contents of configuration properties.

3.53

standard network-variable type

SNVT

network-variable type that has been standardized by the CEA-709.6

3.54

standard program ID

SPID

eight-byte number that uniquely identifies the device interface for a device

NOTE Encoded according to rules specified in clause 4.3.

3.55

static functional block

functional block that is statically defined for a device; that is, a functional block that is not a dynamic functional block

3.56

static network variable

network variable that is statically defined for a device; that is, a network variable that is not a dynamic network variable

3.57

subsystem

two or more devices working together to perform a function and bearing fixed, pre-defined relationships to one another

NOTE A subsystem may use one or more ANSI/CEA-709.1 domains.

3.58

successful commissioning

<noun> process of taking a device and integrating it into a network

<adjective> device can be physically installed in a network and made to perform its application function with the exclusive use of its device interface and a choice of third-party tools

3.59

system

one or more independently managed subsystems working together to perform a function

NOTE A system may use one or more ANSI/CEA-709.1 domains.

3.60

unconfigured device

device without a valid network configuration

3.61

usage

one-byte value describing the intended usage of the device and is part of the SPID of a device

NOTE The usage field consists of a one-bit changeable-interface flag, a one-bit functional-profile-specific flag, and a 6-bit usage ID.

3.62

usage ID

six-bit value in the least-significant portion of the usage field that identifies the primary intended usage of a device

3.63

user data

non-standardised user functional blocks, user network variables, and user configuration properties used by a device manufacturer to augment the device interface

3.64

wink function

function provided by a device that allows a network integrator to physically identify the device

EXAMPLE A wink function may blink an LED on the device.

4 Device Interfaces

4.1 General

The device interface is the network-visible interface to a device. The elements that comprise an interoperable device interface include the unique node ID, the standard program ID, the device channel ID, the device location field, the device self-documentation string, the device configuration properties, and the functional blocks.

Following is a summary of each of the elements:

- *Unique node ID*. A 48-bit unique identifier for a CNP device.
- *Standard program ID (SPID)*. A number that uniquely identifies the device interface for a device.
- *Device channel ID*. A number that optionally specifies the channel to which the device is attached.
- *Device location field*. A string or number that optionally specifies the device location.
- *Device self-documentation string*. A string that specifies the functional blocks on a device.
- *Device configuration properties*. Configuration data used to configure the device. Functional blocks may also have configuration properties.
- *Functional blocks*. Logical components implemented on the device.

With the exception of the unique node ID, a network tool can read all of these elements directly from a device over the network, or from the device interface (XIF) file for the device as described in 4.10 *Device Interface (XIF) File*. The benefit of making this information available directly from the device itself is that a network tool can read all of the information needed to integrate and manage the device over the network, and no accompanying manufacturer documentation is required. The benefit of making this information available from a device information file is that the device may be designed into a network before physical access to the device is available. The latter method is typically used for engineered systems, but the former method is sometimes used when a device interface file is not available.

The device interface elements are described in the remainder of this clause. Device configuration properties are described in 4.8.4.

4.2 Unique Node ID

The unique node ID is a 48-bit number within the read-only data structure of a device as defined by the ANSI/CEA-709.1 protocol. The unique node ID is a unique number written to a processor during development or manufacturing. Network tools use the unique node ID to send network installation messages to a device, prior to the device being assigned a network address as described in 6.2, Network Addressing.

Guideline 4.2: A device shall implement a unique node ID as defined in 4.2, Unique Node ID.

Manufacturers may wish to provide two copies of the unique node ID in a human- or machine-readable format, attached to the product. One copy should be removable so that an installer may place it on a system drawing, or similar plan. This can even be done using a barcode for ease and accuracy of unique node ID input into a network tool. An example unique node ID barcode label is shown in the following figure.



Figure 2 — Example Unique Node ID Barcode Sticker

While this standard does not mandate a bar-coding method, the CODE-39 format (ISO/IEC 16388) SHOULD be used for compatibility with many readily available barcode readers.

4.3 Standard Program ID

4.3.1 General

The *standard program ID* (SPID) is an 8-byte number within the read-only data structure of a device as defined by the ANSI/CEA-709.1 protocol. It uniquely identifies the device interface for a device. It is used by network tools to associate a device with a device interface definition. This speeds up the commissioning process by allowing a network tool to obtain the device interface definition without uploading the entire definition from every device.

4.4 Guideline 4.3: A device shall implement a standard program ID as defined in 4.3, Standard Program ID

General.

The 16 hex digits of the SPID are organized as 6 fields that identify the format (F), manufacturer (M), device class (C), usage (U), channel type (T), and model number (N) of the device. These 6 fields are organized as follows, and are described in the following sections:

FM:MM:MM:CC:CC:UU:TT:NN

The manufacturer, classification, channel type, and optionally the usage fields contain values defined by the CEA-709.6.

4.4.1 Format Field

The Format field contains a four-bit value defining the structure of the program ID and device self-documentation strings. The format shall be “9” for all devices unless specified to be “8” during testing of the device.

4.4.2 Manufacturer Field

The Manufacturer field contains a 20-bit manufacturer ID (MID). The MID uniquely identifies the device manufacturer.

4.4.3 Device Class Field

The Device Class field is a two-byte value identifying the primary function of the device.

This value is drawn from a registry of pre-defined Device Class definitions.

A device may implement multiple functional blocks. One of these functional blocks may be designated as the primary functional block, and the definition of this functional block is called the primary functional profile. If the primary functional profile number is greater than 99 and less than 20 000, the device class may be set to the profile number.

Standard functional profiles are also given device classes equal to their functional profile number. If a developer chooses to use a device class that is assigned to a standard functional profile, then the device containing that device class shall contain a functional block implementation of that profile.

4.4.4 Usage Field

4.4.4.1 General

The Usage field is a one-byte value describing the intended usage of the device. The Usage field consists of a one-bit Changeable-Interface flag, a one-bit Functional Profile-Specific flag, and a 6-bit usage ID. These subfields are described in the following sections.

4.4.4.2 Changeable-Interface Flag

The *Changeable-Interface flag* is the msb of the Usage field. It shall be set if the device uses changeable network variable types or dynamic network variables as described in 4.8.3.3 Changeable-Type Network Variables. The flag shall be clear if the device uses a fully static device interface.

4.4.4.3 Functional Profile-Specific Flag

The *Functional Profile-Specific* flag is the second-msb of the Usage field. It shall be set if the usage ID value is defined by the primary functional profile for the device. The flag shall be clear if the usage ID value is defined by the standard usage ID values or if the Device Class field does not identify the functional profile number of the primary functional profile for the device.

4.4.4.4 Usage ID

The *usage ID* is a 6-bit value in the least-significant portion of the Usage field that identifies the primary intended usage of the device. Based on the setting of the Functional Profile-Specific flag, the usage ID is defined by one of the following:

- If the Functional Profile-Specific flag is clear, the usage ID shall be set to one of the standard usage ID values.
- If the Functional Profile-Specific flag is set, the usage ID shall be set to one of the usage ID values specified by the primary functional profile for the device, as determined by the Device Class field.

4.4.5 Channel Type Field

The Channel Type field is a one-byte value identifying the communications channel type supported by the device's network transceiver. The standard channel-type values are drawn from a registry of pre-defined channel-type definitions.

4.4.6 Model Number Field

The Model Number field is a one-byte value identifying the specific product model of the device. Model numbers are assigned by the product manufacturer and shall be unique within the device class, usage ID, and channel type for a manufacturer ID. The same hardware may be used for multiple model numbers depending on the program that is loaded into the hardware. The model number within the SPID does not have to conform to the manufacturer's marketing or engineering model numbers. It can be used as a decimal reference, hexadecimal reference, or any other method of convenience.

Table 1 — Examples of model number fields

Kind of Model	Numbers
Decimal Model Numbers	0; 1; 2; 3;...9; 10; 11... = sequential by 1 10; 20; 30;...90; 100; 110... = incremental by 10s' place
Hexadecimal Model Numbers	01; 02; 03;...09; 0A; 0B... = sequential by 1 10; 20; 30;...90; A0; B0... = incremental by nibbles' place
ASCII-Character Model Numbers	"A"; "B"; "C"; ... = sequential by 1 using ASCII values for the representation of characters (0x41; 0x42; 0x43;...)

4.5 Device Channel ID

The device channel ID is a 2-byte unsigned-long field within the configuration structure of a device as defined by the ANSI/CEA-709.1 protocol. Network tools may use the device channel ID to track the channel to which a device is attached. A value of zero indicates that the device's channel ID is unassigned. When a device is shipped unconfigured and installed in a managed network, the device application shall not require specific values in this field since a network tool may change the value as needed.

Guideline 4.5A: A device shall be manufactured with a zero channel ID — unless it is shipped configured or is a self-installing device; in which case, the channel ID can be non-zero.

Guideline 4.5B: A device shall not modify its channel ID field — unless it is shipped configured or is a self-installing device; in which case, the device application can modify the channel ID.

4.6 Device Location Field

The device location field is a 6-byte field within the configuration structure of a device as defined by the ANSI/CEA-709.1-protocol. Integrators, network tools, or the device itself may use this field to document the physical location of a device. This field may be read

and written over the network using the Read Memory and Write Memory network-management messages defined by the ANSI/CEA-709.1 protocol. Some devices can determine their physical location by reading external physical inputs such as DIP switches, keyed connectors, or card-cage slot numbers. Such devices may use the device location field to communicate their physical location information to a network tool that can use this information to identify the physical location of the device.

Use of this field is optional, but if used shall conform to the following guideline. A device may optionally implement a location configuration property (see 4.8.3.5 Dynamic Network Variables and Functional Blocks) that can be used to provide a more complete location description than is possible in the 6-byte location field. The location configuration-property value is a string of up to 31 characters. When used for device location, the location configuration property shall apply to the Node Object functional block of the device if the device has a Node Object functional block, otherwise it shall apply to the entire device. If a device has multiple locations, such as a device with multiple remote sensors, each of the functional blocks on the device may also implement a location configuration property to identify the location of each of the remote components. The location configuration property associated with the Node Object identifies the location of the device itself, whereas the other location configuration properties identify the locations of their respective hardware components.

Guideline 4.6A: A device's application program that wishes to communicate its physical location or ID assignment to a network tool can write this information into the location ID field of its configuration structure when the device is reset. If the most-significant bit of the first byte is one, the information is encoded as a 15-bit unsigned integer in the range 0 to 32 767, with the most-significant 7-bits in the lower 7-bits of the first byte (location[0]). If the most-significant bit of the first byte is zero, the information is encoded as a 0- to 6-character ASCII string. If the string is shorter than six characters, it shall be null-terminated (0x00).

Guideline 4.6B: A device that implements a location configuration property to represent the device location shall apply the CP to the Node Object functional block if the device has a Node Object functional block, and shall otherwise apply the CP to the entire device.

4.7 Device Self-Documentation String (DSDS)

The device self-documentation string is a string of up to 1024 bytes (subject to device memory limits) within the self-identification structure of a device as defined by the ANSI/CEA-709.1 protocol. This string specifies the self-documentation string structure, the functional blocks, and optionally describes the function of a device.

Guideline 4.7A: A device shall contain a device self-documentation string that specifies the self-documentation string structure and the functional profiles implemented by each functional block on the device as described in 4.7, Device Self-Documentation String.

Guideline 4.7B: A device shall store the device self-documentation string in the application image as described in the ANSI/CEA-709.1 protocol.

The syntax for the device self-documentation string without arrays is as follows:

```
&3.4@Functionalblock,FunctionalBlock;selfDocText
```

The syntax for the device self-documentation string with arrays is as follows: