TA Document 1999027
Configuration ROM for AV/C Devices 1.0

December 12, 2000

Sponsored by:
1394 Trade Association

Accepted for Release by:
1394 Trade Association Board of Directors.

Abstract:
The configuration ROM for AVC Devices specification describes required entries and their order in configuration ROM space.

Keywords:
Configuration ROM, Unit Directory, Bus information block, Driver identification

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
Regency Plaza Suite 350
2350 Mission College Blvd.
Santa Clara, Calif. 95054, USA

This page is left intentionally blank

Table of Contents

List of Figures

List of Tables

1. Overview

1.1 Purpose

The purpose of this specification is to clarify and to provide rules for using configuration ROM space to support the needs for AV/C devices. The new rules are for the purposes of providing known and expected order in ROM which ultimately leads to better interoperability and support for user interfaces.

1.2 Scope

This document confines itself only to the configuration ROM space as it applies to AV/C devices. This document builds upon IEEE P1212 Draft 1.0, “Draft Standard for a Control and Status Registers (CSR) Architecture for microcomputer buses” section 7[R4], and IEEE Std 1394a-2000 “Standard for a High Performance Serial Bus – Amendment 1” section 10.9, 10.25 and 10.26[R2], and provides the data structures necessary to support an enhanced user interface with textual and icon information, and the ability for a controlling device to determine the appropriate driver software for the AV/C device. Furthermore, guidelines for data ordering in the configuration ROM space have been introduced to ensure interoperability between devices, and mandatory and optional entries have been defined.

2. References

The following standards contain provisions, which through reference in this document constitute provisions of this standard. All the standards listed are normative references. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

[R1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus.

[R2] IEEE Std 1394a-2000, IEEE Standard for a High Performance Serial Bus – Amendment 1.

[R3] ISO/IEC13213, Microprocessor systems – Control and Status Registers (CSR) Architecture for microcomputer buses.

[R4] IEEE P1212 Draft 1.0, Draft Standard for a Control and Status Registers (CSR) Architecture for microcomputer buses.

[R5] IEC 61883-1, Consumer audio/video equipment – Digital interface – Part 1: General.

[R6] AV/C Digital Interface Command Set General Specification, Version 3.0. TA document number 1998003.

[R7] 1394 TA Technical Bulletin TB001, CRC Verification for Configuration ROM

3. Definitions

3.1 Conformance levels

3.1.1 expected: A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Specification. Other hardware and software design models may also be implemented.

3.1.2 may: A key word that indicates flexibility of choice with *no implied preference*.

3.1.3 shall: A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.

3.1.4 should: A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

3.1.5 reserved fields: A keyword used to describe objects–bits, bytes, quadlets, octlets and fields–or the code values assigned to these objects in cases where either the object or the code value is set aside for future standardization. Usage and interpretation may be specified by future extensions to this or other standards. A reserved object shall be zeroed or, upon development of a future standard, set to a value specified by such a standard. The recipient of a reserved object shall not check its value. The recipient of an object whose code values are defined by this standard shall check its value and reject reserved code values.

3.2 Glossary of terms

3.2.1 AV/C Device: An AV device that implements the AV/C protocol as defined by the AV/C Digital Interface Command Set General Specification and subunit type specifications.

3.2.2 byte: Eight bits of data.

3.2.3 CSR Architecture: A convenient abbreviation of the following reference (see clause 2): IEEE P1212 Draft 1.0, Draft Standard for a Control and Status Register (CSR) Architecture for microcomputer buses[R4].

3.2.4 quadlet: Four bytes of data.

3.2.5 Minimal ASCII subset: The minimal ASCII subset is defined in section 7.4 of IEEE P1212[R4]. This subset is derived from ISO/IEC646:1991.

3.3 Acronyms and abbreviations

AV/C Audio Video Control

ROM Read Only Memory

MSB Most Significant Byte

LSB Least Significant Byte

4. ROM Formats

The following sections provide information about various ROM formats as they apply to AV/C devices.

4.1 Configuration ROM during power reset initialization

During the initialization process that follows a power reset, an AV/C device may not be able to respond immediately to Serial Bus request subactions addressed to parts of its configuration ROM. When a device has not yet made a configuration ROM accessible, it shall return a data value of zero for read requests addressed to its first quadlet (FFFF F000 0400_{16}), as specified in section 7.2 of IEEE P1212[R4] or shall acknowledge the request subaction with *ack_tardy*, as specified in section 10.9 of IEEE Std 1394a-2000 [R2]. Once power reset initialization completes, an AV/C device shall return a nonzero data value for read requests addressed to the first quadlet of a configuration ROM; this indicates that the remainder of the device's configuration ROM may be read.

4.2 Unused area

An AV/C device shall not return "address error" for quadlet read requests addressed to between FFFF F000 0400_{16} and FFFF F000 $07FF_{16}$ inclusive. Some of the locations within the first kilobyte of configuration ROM may be unused in the sense that they are not part of the navigable hierarchy of configuration ROM; these locations shall return response data values of zero.

4.3 General ROM format

An AV/C device shall implement the general format configuration ROM in accordance with IEEE P1212[R4] and IEEE Std 1394a-2000[R2].

A general format configuration ROM has a bus information block, a root directory, and optional subsidiary directories and/or leaves as shown in Figure 4.1.

AV/C devices should set the *crc_length* equal to the *bus_info_length*, whose value is 04_{16} in IEEE Std 1394a-2000.

NOTE —More detailed information about the *bus_info_length* and the *crc_length* fields are defined in section 7.2 of IEEE P1212[R4].

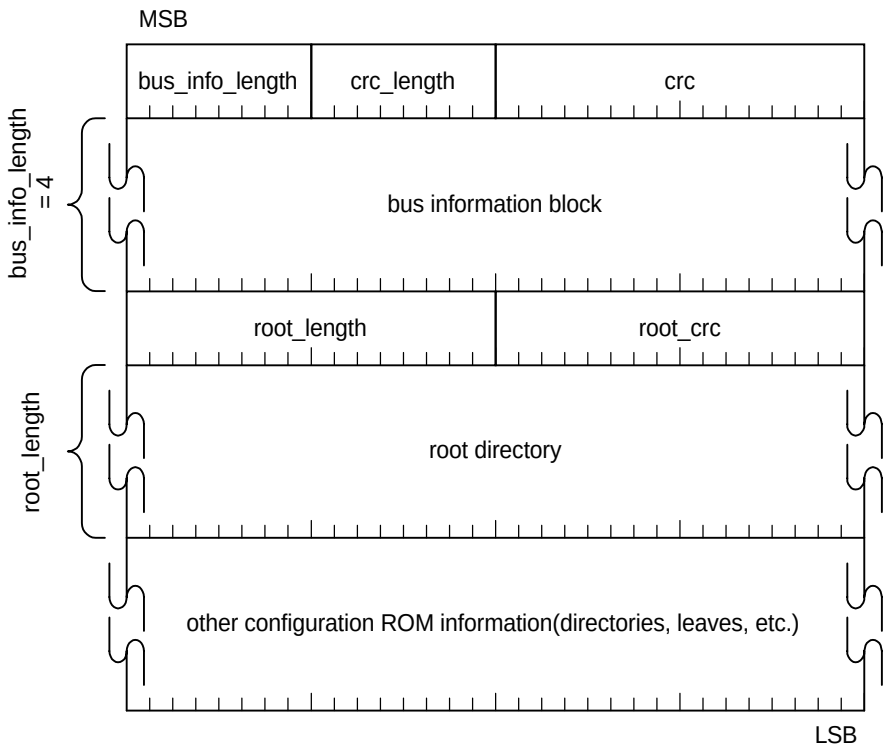

Figure 4.1 – General ROM format

4.4 CRC calculation

Data structures defined in this standard provide CRCs that may be used to detect data errors when reading configuration ROM. The CRC calculation is based on the ITU-T CRC-16 code (ITU-T Recommendation V.41).

The calculation method is defined in section 7.3 of IEEE P1212[R4].

There is a special recommendation for the configuration ROM reader with regard to legacy devices. Please refer to TB001[R7].

4.5 Data structures

The configuration ROM has a hierarchical structure, which is based on elemental data such as directories, leaves, and/or descriptors.

NOTE − More detailed information about data structures is defined in section 7.5 of IEEE P1212[R4].

4.6 Bus information block

The format of the bus information block defined in section 10.25 of IEEE Std 1394a-2000[R2] is shown in the figure below.

Figure 4.2 – Bus information block

An AV/C device shall meet these following requirements:

— The *irmc* bit shall be one (isochronous resource manager capable).

— The *cmc* bit shall be one (cycle master capable).

— It is recommended that an AV/C device implement the *max_ROM* value of one or two (at least block read requests aligned on 64-byte addresses with a data length of 64 bytes are supported). If *max_ROM* value is zero, the configuration ROM should be read with quadlet aligned quadlet read transaction.

NOTE − These fields and bits are defined in section of 8.3.2.5.5.1 IEEE Std 1394-1995[R1] and section 10.25 of IEEE Std 1394a-2000[R2].

5. Directory entries

5.1 Root directory

5.1.1 Required entries and their order

The following entries and groups of entries are required in the root directory and their order is recommended.

— Vendor_ID entry, and its textual descriptor entry for a descriptor in minimal ASCII subset

— Model_ID entry, and its textual descriptor entry for a descriptor in minimal ASCII subset

— Node_Capabilities entry

— Unit_Directory entry for AV/C protocol (see note below)

NOTE — It is recommended that configuration ROM designers include the AV/C Unit_Directory entry in the root directory when support of legacy controllers is required (legacy controllers contain discovery software that predates the definition of instance directories). Please see section 5.2 of this standard: Instance directory for more information.

Optional entries may be placed between or after these entries. The Vendor_ID and the Model_ID entries may have textual descriptor(s) in other character set(s) and icon descriptor(s).

Vendor_ID, Model_ID and Node_Capabilities entries are defined in the section 7.7 of IEEE P1212[R4].

Figure 5.1 shows an example of the root directory for AV/C devices.

root_length	crc
03_{16}	vendor_ID
81_{16} or $C1_{16}$	descriptor leaf or directory offset
17_{16}	model_ID
81_{16} or $C1_{16}$	descriptor leaf or directory offset
$0C_{16}$	node_capabilities
$D1_{16}$	unit directory offset(AV/C protocol)
additional fields if necessary	
……	

Figure 5.1 – Entries of the root directory

5.1.2 Vendor_ID and its descriptor(s)

A textual descriptor in the minimal ASCII subset is required for describing the Vendor_ID. Textual descriptors in other character sets and icon descriptors are optional.

Table 5.1 – Descriptor types and support levels

Descriptor Type	Support level
Textual descriptor in minimal ASCII subset	Mandatory
Textual descriptors in other character sets	Optional
Icon descriptor in the YCbCr (48x48) format	Optional (Mandatory, if any icon descriptor is present)
Icon descriptors in other icon formats	Optional

When only the minimal ASCII subset descriptor is present, the descriptor entry shall immediately follow the Vendor_ID, as shown in Figure 5.2(a).

When other textual descriptor(s) and/or icon descriptor(s) are present, a descriptor directory shall immediately follow the Vendor_ID, the entry for the minimal ASCII subset descriptor shall be present in the descriptor directory. The order of the entries for the descriptors is recommended as below and shown in Figure 5.2(b);

— Textual descriptor in minimal ASCII subset

— Icon descriptor in the YCbCr (48x48) format

— Textual descriptors in other character sets

— Icon descriptors in other icon formats

NOTE —If any icon descriptor is present, the YCbCr (48x48) format is required for the Vendor_ID. For more information about the icon descriptor in the YCbCr (48x48) format, refer to section 5.4 of this specification.

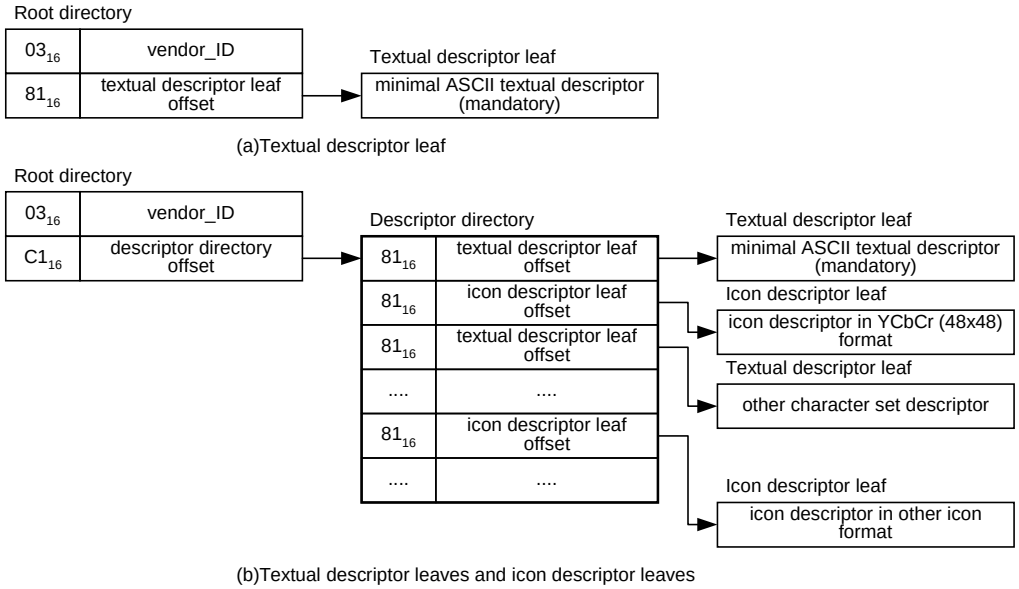

Figure 5.2 – Vendor_ID and its descriptor(s)

5.1.3 Model_ID and its descriptor(s)

A textual descriptor in the minimal ASCII subset is required for describing the Model_ID. Textual descriptors in other character sets, modifiable textual descriptors, icon descriptors and modifiable icon descriptors are optional.

NOTE − More detailed information about these descriptors is defined in section 7.5.4 of IEEE P1212[R4].

Table 5.2 – Descriptor types and support levels

Descriptor Type	Support level
Textual descriptor in minimal ASCII subset	Mandatory
Textual descriptors in other character sets	Optional
Modifiable textual descriptors	Optional
Icon descriptor in the YCbCr (48x48) format	Optional (Mandatory, if any icon descriptor is present)
Icon descriptors in other icon formats	Optional
Modifiable icon descriptors	Optional

When only the minimal ASCII subset descriptor is present, the descriptor entry shall immediately follow the Model_ID, as shown in Figure 5.3(a).

When other textual descriptor(s) and/or icon descriptor(s) are present, a descriptor directory shall immediately follow the Model_ID, the entry for the minimal ASCII subset descriptor shall be present in the descriptor directory. The order of the entries for the descriptors is recommended as below and shown in Figure 5.3(b);

- Textual descriptor in minimal ASCII subset
- Icon descriptor in the YCbCr (48x48) format
- Textual descriptors in other character sets
- Modifiable textual descriptors
- Icon descriptors in other icon formats
- Modifiable icon descriptors

NOTE − If any icon descriptor is present, the YCbCr (48x48) format is required for the Model_ID. For more information about the icon descriptor in the YCbCr (48x48) format, refer to section 5.4 of this specification.

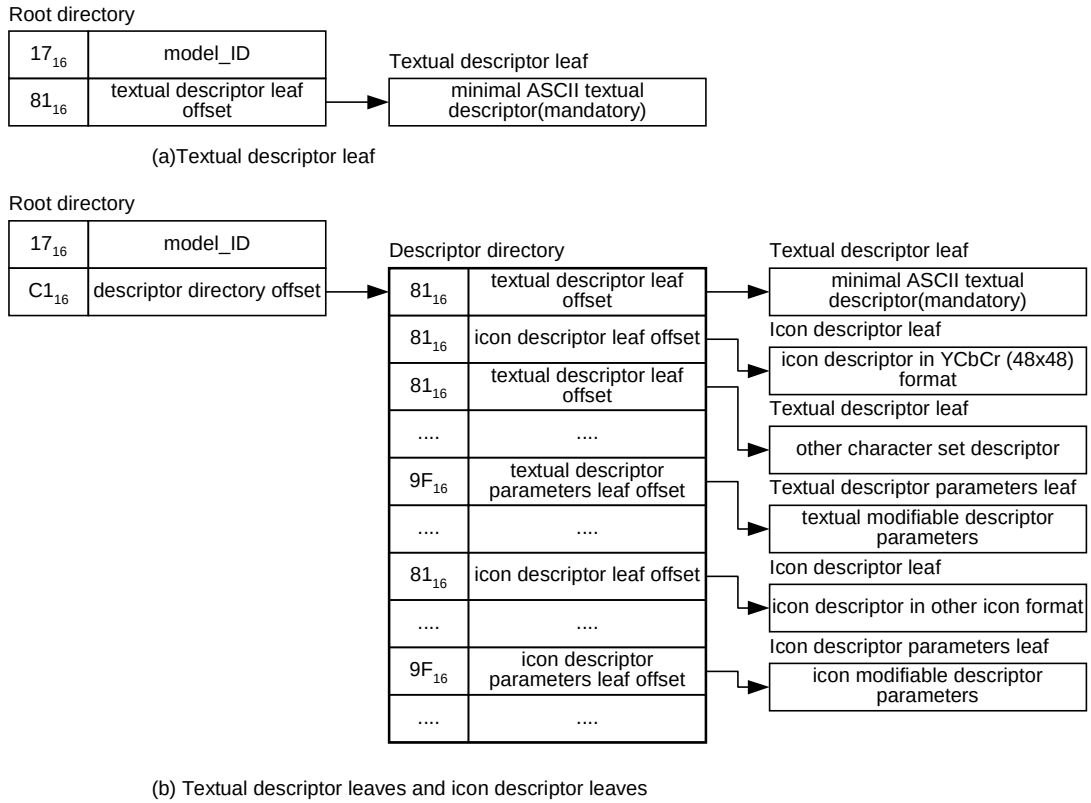

Figure 5.3 – Model_ID and its descriptor(s)

5.1.4 Node_Capabilities entry

The *node_capabilities* field contains subfields specified by ISO/IEC 13213:1994[R3]. AV/C devices shall implement the SPLIT_TIMEOUT register, the 64-bit fixed addressing scheme, the STATE_CLEAR.*lost* bit and the STATE_CLEAR.*dreq* bit , which are indicated by setting the *spt*, *64*, *fix*, *lst* and *drq* bits to one.

If no other *node_capabilities* bits are one, this results in a value of $0083C0_{16.}$

5.1.5 Unit_Directory entry for AV/C protocol

The Unit_Directory entry for AV/C protocol is recommended to implement in the root directory to support legacy readers.

5.1.6 Other entries

5.1.6.1 Node_Unique_ID entry

The Node_Unique_ID entry which is required in IEC61883-1[R5], is removed for AV/C devices that implement this specification. Legacy devices may contain this entry. The same information is contained in the bus information block, from where a device can obtain the information if necessary.

NOTE — Please refer to the table 12 of IEEE P1212[R4] regarding the use of the Node_Unique_ID.

5.1.6.2 Instance_Directory entry

Instance_Directory entries shall be present in the root directory if instance directories are present.

5.2 Instance directory

Instance directories and their associated keyword leaves are not mandated by this specification, except when there is more than one unit directory in the configuration ROM, in which case they shall be implemented in accordance with IEEE P1212[R4]

When an Instance_Directory entry(s) is present in the root directory, at least one of the instance directories shall contain a Unit_Directory entry that points to an AV/C protocol.

NOTE — One of the primary uses of configuration ROM is the efficient discovery of device functions expressed within the node. Instance directories provide a structure to describe particular instantiations of functions. It is recommended that controllers search the instance directory structure for functionality before searching unit directories included in the root directory. It is also recommended that configuration ROM designers use instance directories to describe functionality within the node. Further discussion of instance directories can be found in IEEE P1212[R4]

5.3 Unit directory for the AV/C protocol

5.3.1 Entries and their order in the unit directory for AV/C

Specifier_ID and Version are required and Model_ID and its descriptor entry are recommended in the unit directory, which specifies the AV/C protocol. These entries, if present, shall be placed in the following order.

— Specifier_ID

— Version

— Model_ID and its descriptor entry

Optional entries may be placed after these entries, but no entries shall be placed between “Specifier_ID and Version” and “Model_ID and its descriptor entry”.

Figure 5.4 below shows an example of unit directory.

unit_directory_length	crc
12_{16}	specifier_ID ($00A02D_{16}$)
13_{16}	version (010001_{16})
17_{16}	model_ID
81_{16}	textual descriptor leaf offset(minimal ASCII subset)
additional field(s) if necessary	
……	

Figure 5.4 – Entries of the Unit directory

5.3.2 Specifier_ID and Version

The *specifier_ID* shall be $00A02D_{16}$ and the *version* shall be 010001_{16} for AV/C devices.

NOTE —More detailed information about the Specifier_ID and the Version entries is provided in section 7.7.10 and 7.7.11 of IEEE P1212[R4].

5.3.3 Model_ID and its descriptor

If Model_ID entry is present in the unit directory, the textual descriptor in the minimal ASCII subset is required for the Model_ID entry in the unit directory for the AV/C protocol, as shown in Figure 5.5. The Model_ID shall not have any other descriptor. Legacy devices developed prior to this specification may not have these entries.

Some controllers use the Model_ID and its descriptor to identify the appropriate software.

Some legacy controllers may not expect the model_ID and its descriptor to be in the unit directory. When support of these legacy controllers is required, the model_ID and its descriptor may be omitted from the unit direcotry.

Table 5.3 – Descriptor type and support level

Descriptor Type	Support level
Textual descriptor in minimal ASCII subset	Mandatory

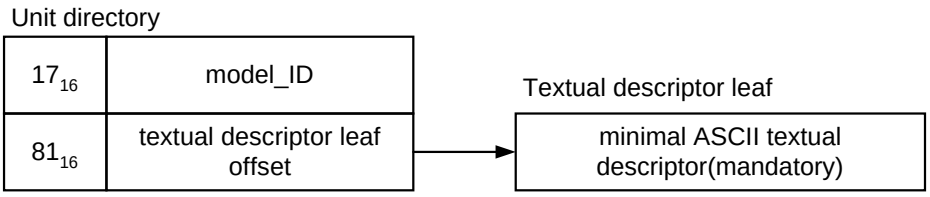

Figure 5.5 – Model_ID and its descriptor in the unit directory

5.4 Icon descriptor for AV/C devices

The format of an icon descriptor leaf, illustrated by Figure 5.6, is specified in IEEE P1212[R4]. Please also refer to IEEE P1212[R4] for the definition of all parameters.

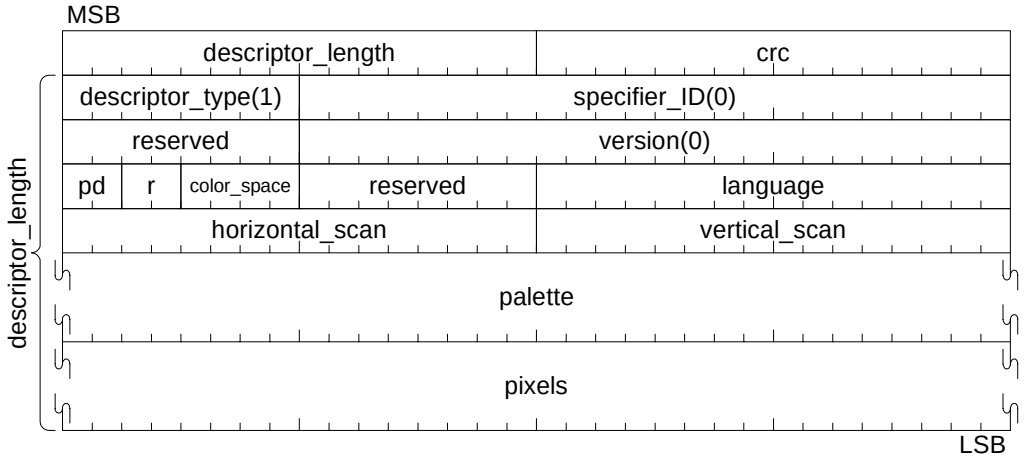

Figure 5.6 – Icon descriptor leaf format

5.4.1 Icon descriptor in YCbCr (48x48) format

Figure 5.7 illustrates an example of icon descriptor with the YCbCr (48x48) format, its pixel size is fixed to 48 by 48, and with no palette data.

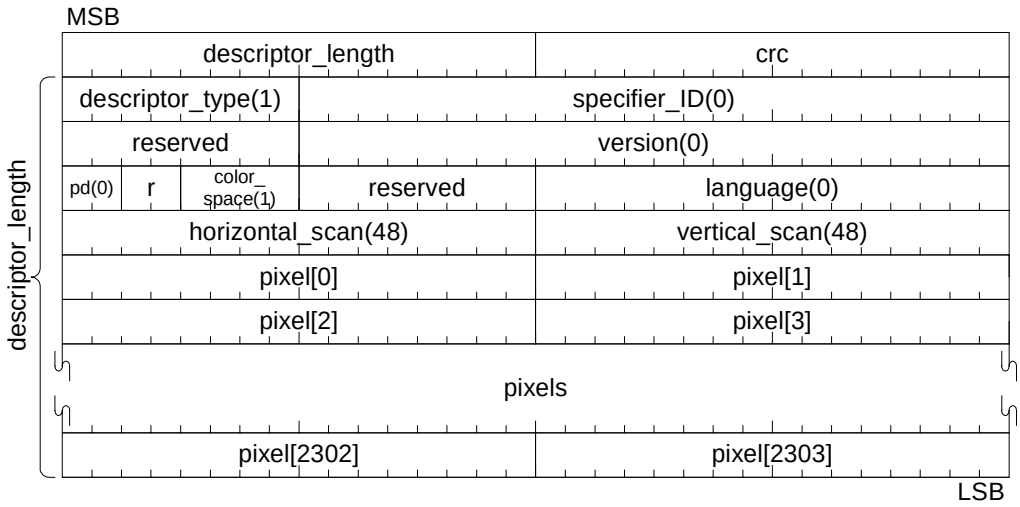

Figure 5.7 – An example icon descriptor in the YCbCr (48x48) format

The *descriptor_type* field shall have the value of one and the *specifier_ID* and *version* fields shall have the value of zero.

The *pd* bit shall have the value of zero, indicating there is no palette. The *color_space* field shall have the value of one, which indicates the YCbCr format.

The *language* field shall have the value of zero, which implies that no language information is specified.

The *horizontal_scan* field shall have the value of 48, and the *vertical_scan* field shall also have the value of 48.

The remainder of the icon descriptor is occupied by the pixels[x] fields that form the image. Each pixel is encoded in a packed YCbCr format and the order of pixels shall be in accordance with IEEE P1212[R4]

Annexes

Annex A:Consideration for configuration ROM reader design (informative)

A.1 Vendor directory

This section describes the information of the Model_ID and its descriptor in the vendor directory of a legacy device.

Some legacy devices may have their Model_IDs and textual descriptors in minimal ASCII subset in their vendor directories.

It is recommended for a controller to read the Model_ID and its descriptor in the vendor directory of a target only if the Model_ID and its descriptor are not present in the root directory of the target.

NOTE —The Model_ID and its descriptor in the vendor directory are intended for supporting the user interface, not for determining driver identification.

Figure A.1 below shows an example of the configuration ROM data structure of Model_ID and its descriptor in the vendor directory.

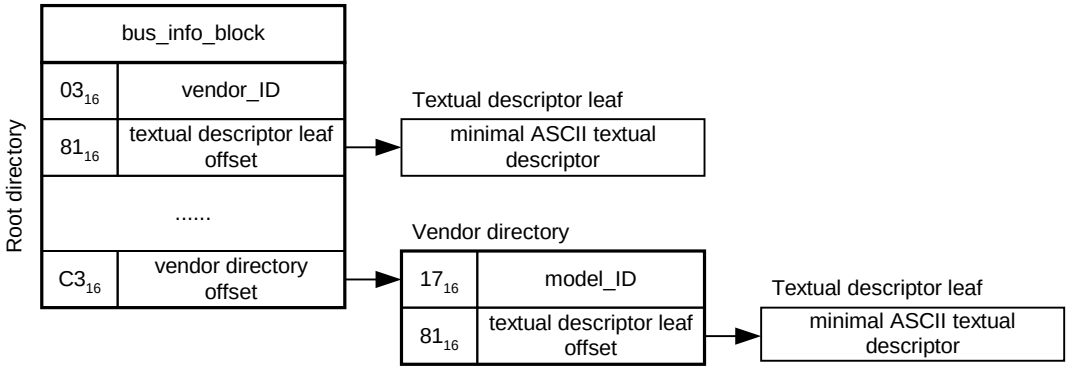

Figure A.1 – Model_ID and its descriptor in the vendor directory

A.2 Configuration ROM read operation

This section describes the information of a configuration ROM read operation.

When a configuration ROM reader reads the configuration ROM, the reader first reads the quadlet located at FFFF F0000 0400_{16} with a quadlet read. If non-zero value is successfully read from the quadlet, then the configuration ROM can be read with block reads or quadlet reads.

Annex B:Module_info directory(informative)

This section describes the usage of the Module_Info directory and Module_Primary_EUI_64 entries in the configuration ROM. The Module_Info directory and the Module_Primary_EUI_64 entries are optional.

When the module has more than one node, the configuration ROM of the primary node may have the Module_Info directory in the root directory in addition to the other entries, directories and descriptors as required in this specification. The configuration ROMs of the other nodes may have the Module_Primary_EUI_64 entry in the root directory to indicate the primary node.

When the Module_Info directory is present, as a minimum, it describes the module using the following entries:

1) Vendor_ID entry, and its descriptor entry for a descriptor in minimal ASCII subset
2) Model_ID entry, and its descriptor entry for a descriptor in minimal ASCII subset

For more information about optional entries, refer to section 5.1.2 and 5.1.3.

When the Module_Primary_EUI_64 entry is present, it contains the eui_64 of the primary node.

NOTE − More detailed information about Module_Info and Module_Primary_EUI_64 is defined in section 7.7.5 and 7.7.6 of IEEE P1212[R4].

Figure B-1 a, b) show an example of a configuration ROM where the module has three nodes.

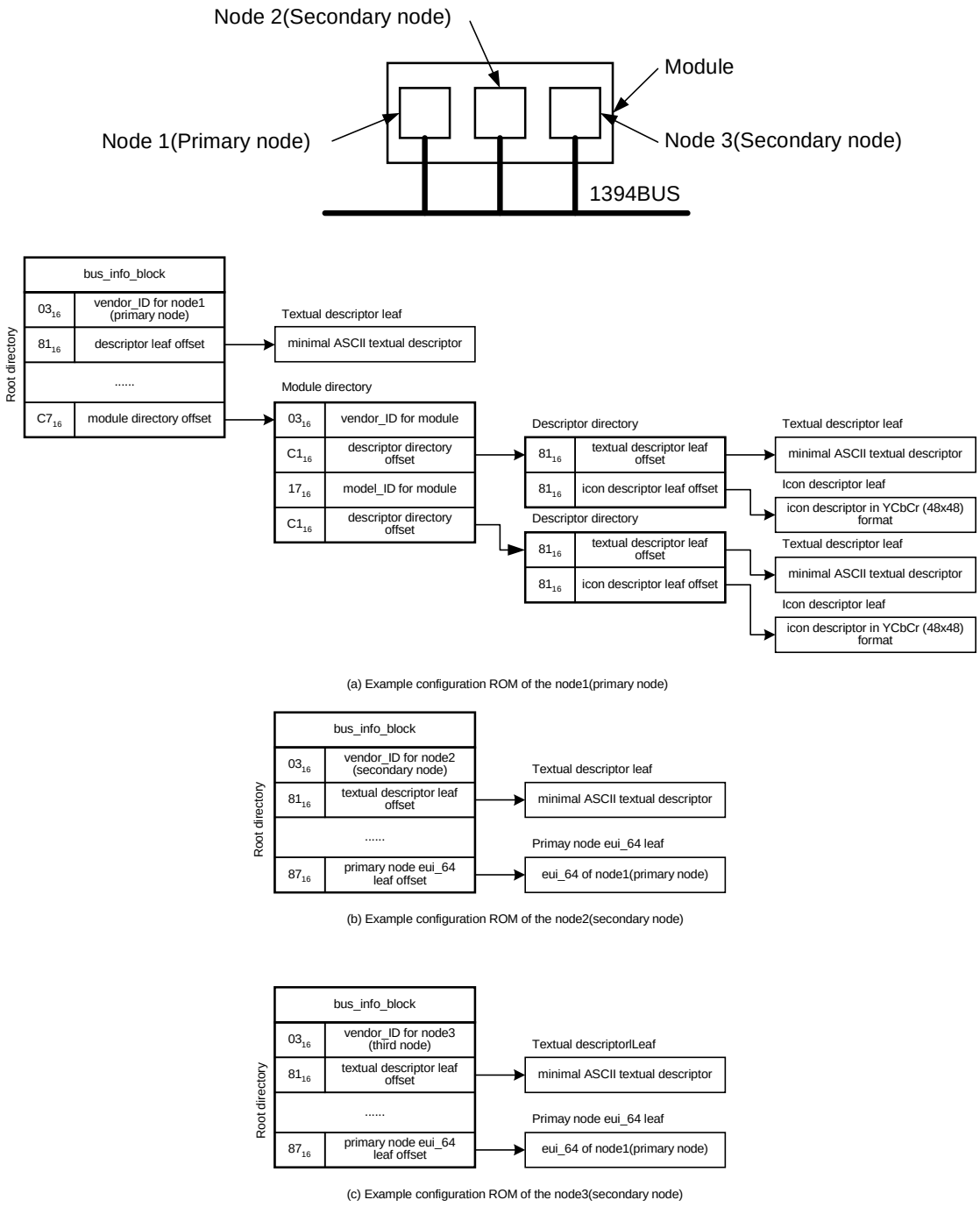

Figure B.1 – Configuration ROM example for a module that has three nodes

In some cases, Module_Info directory may be present, even if the module has one node. For example, separate module information, such as module vender and module model name, may be desired in addition to the node information. Another example is the case where only one node is installed to a module which can have more than one node.

Figure B-2 shows an example of a configuration ROM where a module has one node.

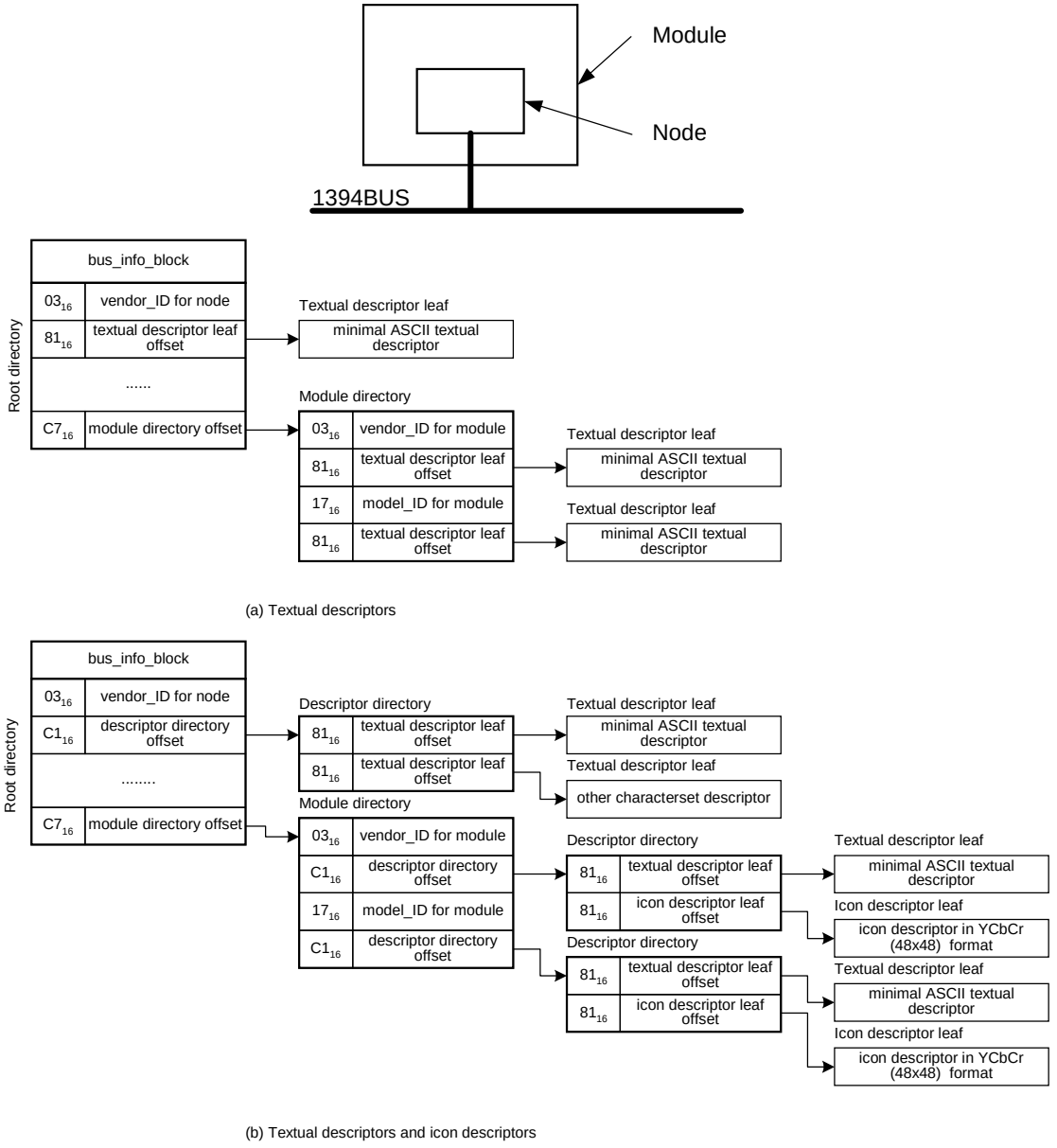

Figure B.2 – Configuration ROM example for a module that has one node

Annex C:Configuration ROM example(informative)

C.1 Simple AV/C device

The figure below illustrates an example of a configuration ROM for AV/C device.

NOTE − The values $FFFFFF_{16}$ for the *vendor_ID*, the *model_ID*, and the eui_64 are for example only. Do not use these values in a real device.

unit :hexadecimal
Rom Base Address :$FFFF\ F000\ 0400_{16}$

1Quadlet = 4Byte / entry field

					value of data				address
bus info block	bus_info_length	crc_length	crc		04	04	EA	BF	0400
	"1"	"3"	"9"	"4"	31	33	39	34	0404
	irmc cmc isc bmc pmc reserved	cyc_clk_acc	max_rec	r max_ROM generation r link_spd	E0	64	61	02	0408
	node_vendor_ID		chip_ID_hi		FF	FF	FF	FF	040C
	chip_ID_lo				FF	FF	FF	FF	0410
root directory	directory_length	crc			00	06	32	87	0414
	03	vendor_ID			03	FF	FF	FF	0418
	81	textual descriptor leaf offset			81	00	00	0A	041C
	17	model_ID			17	FF	FF	FF	0420
	81	textual descriptor leaf offset			81	00	00	0E	0424
	0C	node_capabilities			0C	00	83	C0	0428
	D1	unit directory offset			D1	00	00	01	042C
unit directory	directory_length	crc			00	04	44	2D	0430
	12	specifier_ID			12	00	A0	2D	0434
	13	version			13	01	00	01	0438
	17	model_ID			17	FF	FF	FF	043C
	81	textual descriptor leaf offset			81	00	00	07	0440
descriptor leaf for vendor name	leaf_length	crc			00	05	C9	15	0444
	descriptor_type	specifier_ID			00	00	00	00	0448
	width	character_set	language		00	00	00	00	044C
	"V"	"e"	"n"	"d"	56	65	6E	64	0450
	"o"	"r"	" "	"N"	6F	72	20	4E	0454
	"a"	"m"	"e"	Null	61	6D	65	00	0458
descriptor leaf for model name	leaf_length	crc			00	05	7F	16	045C
	descriptor_type	specifier_ID			00	00	00	00	0460
	width	character_set	language		00	00	00	00	0464
	"M"	"o"	"d"	"e"	4D	6F	64	65	0468
	"l"	" "	"N"	"a"	6C	20	4E	61	046C
	"m"	"e"	Null	Null	6D	65	00	00	0470

Figure C.1 – Example of configuration ROM for simple AV/C device

C.2 Simple AV/C device with modifiable descriptor

The figure below illustrates an example of a configuration ROM for AV/C device that includes a modifiable textual descriptor for the Model_ID.

NOTE —The values $FFFFFF_{16}$ for the *vendor_ID*, the *model_ID*, and eui_64 are for example only. Do not use these values in a real device.

unit :hexadecimal
Rom Base Address :FFFF F000 0400$_{16}$

1Quadlet = 4Byte / entry field

					value of data				address
bus info block	bus_info_length	crc_length	crc		04	04	EA	BF	0400
	"1"	"3"	"9"	"4"	31	33	39	34	0404
	irmc cmc isc bmc pmc reserved	cyc_clk_acc	max_rec	r max_ROM generation r link_spd	E0	64	61	02	0408
	node_vendor_ID		chip_ID_hi		FF	FF	FF	FF	040C
	chip_ID_lo				FF	FF	FF	FF	0410
root directory	directory_length		crc		00	06	C7	E1	0414
	03	vendor_ID			03	FF	FF	FF	0418
	81	textual descriptor leaf offset			81	00	00	0D	041C
	17	model_ID			17	FF	FF	FF	0420
	C1	textual descriptor directory offset			C1	00	00	08	0424
	0C	node_capabilities			0C	00	83	C0	0428
	D1	unit directory offset			D1	00	00	01	042C
unit directory	directory_length		crc		00	04	95	80	0430
	12	specifier_ID			12	00	A0	2D	0434
	13	version			13	01	00	01	0438
	17	model_ID			17	FF	FF	FF	043C
	81	descriptor leaf offset			81	00	00	0A	0440
desc. dir. for model name	directory_length		crc		00	02	7B	A3	0444
	81	descriptor leaf offset			81	00	00	08	0448
	9F	descriptor parameters leaf offset			9F	00	00	0D	044C
descriptor leaf for vendor name	leaf_length		crc		00	05	C9	15	0450
	descriptor_type	specifier_ID			00	00	00	00	0454
	width	character_set	language		00	00	00	00	0458
	"V"	"e"	"n"	"d"	56	65	6E	64	045C
	"o"	"r"	" "	"N"	6F	72	20	4E	0460
	"a"	"m"	"e"	Null	61	6D	65	00	0464
descriptor leaf for model name	leaf_length		crc		00	05	7F	16	0468
	descriptor_type	specifier_ID			00	00	00	00	046C
	width	character_set	language		00	00	00	00	0470
	"M"	"o"	"d"	"e"	4D	6F	64	65	0474
	"l"	" "	"N"	"a"	6C	20	4E	61	0478
	"m"	"e"	Null	Null	6D	65	00	00	047C
descriptor parameters leaf	leaf_length		crc		00	02	07	48	0480
	max_descriptor_size		descriptor_address_hi		00	06	FF	FF	0484
	descriptor_address_lo				F0	01	00	00	0488

modifiable descriptor leaf	leaf_length		crc		00	05	99	B3	10000
	descriptor_type	specifier_ID			00	00	00	00	10004
	width	character_set	language		00	00	00	00	10008
	"M"	"y"	" "	"M"	4D	79	20	4D	1000C
	"o"	"d"	"e"	"l"	6F	64	65	6C	10010
	Null	Null	Null	Null	00	00	00	00	10014

Figure C.2 – Example of configuration ROM for simple AV/C device with modifiable descriptor

C.3 AV/C device with multiple protocols

The figure below illustrates an example of a configuration ROM for AV/C device that includes multiple unit directories.

NOTE —The values FFFFFF$_{16}$ for the *vendor_ID*, the *model_ID*, and eui_64 are for example only. Do not use these values in a real device.

unit :hexadecimal
Rom Base Address :FFFF F000 0400$_{16}$

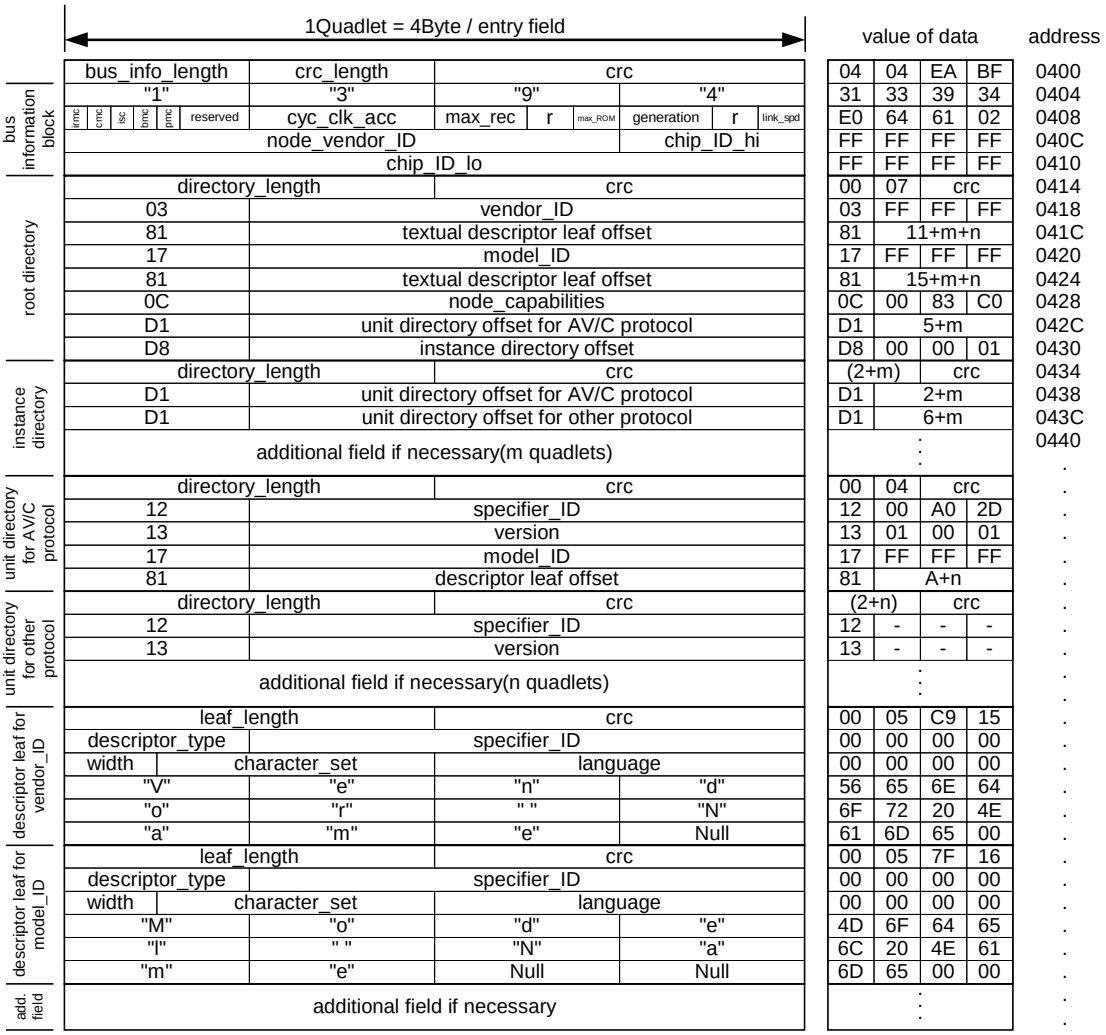

Figure C.3 – Example of configuration ROM for AV/C device with multiple protocols