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

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.
# Table of Contents

1. Overview .................................................................................................................... 7  
   1.1 Purpose .................................................................................................................. 7  
   1.2 Scope .................................................................................................................... 7  

2. References .................................................................................................................. 8  

3. Definitions ................................................................................................................. 9  
   3.1 Conformance levels ............................................................................................... 9  
   3.2 Glossary of terms ................................................................................................. 9  
   3.3 Acronyms and abbreviations ............................................................................... 9  

4. ROM Formats ............................................................................................................. 10  
   4.1 Configuration ROM during power reset initialization ........................................ 10  
   4.2 Unused area .......................................................................................................... 10  
   4.3 General ROM format ......................................................................................... 10  
   4.4 CRC calculation ................................................................................................. 11  
   4.5 Data structures ................................................................................................... 11  
   4.6 Bus information block ....................................................................................... 11  

5. Directory entries ...................................................................................................... 13  
   5.1 Root directory ..................................................................................................... 13  
      5.1.1 Required entries and their order ................................................................. 13  
      5.1.2 Vendor_ID and its descriptor(s) ............................................................... 13  
      5.1.3 Model_ID and its descriptor(s) ................................................................. 15  
      5.1.4 Node_Capabilities entry ...................................................................... 16  
      5.1.5 Unit_Directory entry for AV/C protocol ............................................... 16  
      5.1.6 Other entries ......................................................................................... 17  
   5.2 Instance directory ............................................................................................... 17  
   5.3 Unit directory for the AV/C protocol ................................................................. 17  
      5.3.1 Entries and their order in the unit directory for AV/C ................................ 17  
      5.3.2 Specifier_ID and Version ..................................................................... 17  
      5.3.3 Model_ID and its descriptor(s) ............................................................ 17  
   5.4 Icon descriptor for AV/C devices ......................................................................... 19  
      5.4.1 Icon descriptor in YCbCr (48x48) format ........................................... 19  

Annex A: Consideration for configuration ROM reader design (informative) .................. 21  
   A.1 Vendor directory ............................................................................................... 21  
   A.2 Configuration ROM read operation ................................................................. 21  

Annex B: Module_info directory(informative) .............................................................. 22  

Annex C: Configuration ROM example(informative) .................................................. 25  
   C.1 Simple AV/C device ....................................................................................... 25  
   C.2 Simple AV/C device with modifiable descriptor ........................................... 26  
   C.3 AV/C device with multiple protocols ............................................................ 28
## List of Figures

- Figure 4.1 – General ROM format .................................................................................................................. 11
- Figure 4.2 – Bus information block .............................................................................................................. 12
- Figure 5.1 – Entries of the root directory ...................................................................................................... 13
- Figure 5.2 – Vendor ID and its descriptor(s) ................................................................................................... 14
- Figure 5.3 – Model ID and its descriptor(s) ..................................................................................................... 16
- Figure 5.4 – Entries of the Unit directory ....................................................................................................... 18
- Figure 5.5 – Model ID and its descriptor in the unit directory .......................................................................... 18
- Figure 5.6 – Icon descriptor leaf format ....................................................................................................... 19
- Figure 5.7 – An example icon descriptor in the YCbCr (48x48) format ........................................................... 20
- Figure A.1 – Model ID and its descriptor in the vendor directory .................................................................. 21
- Figure B.1 – Configuration ROM example for a module that has three nodes ........................................... 23
- Figure B.2 – Configuration ROM example for a module that has one node .................................................. 24
- Figure C.1 – Example of configuration ROM for simple AV/C device ........................................................ 25
- Figure C.2 – Example of configuration ROM for simple AV/C device with modifiable descriptor ............ 27
- Figure C.3 – Example of configuration ROM for AV/C device with multiple protocols .......................... 28
List of Tables

Table 5.1 – Descriptor types and support levels ................................................................. 14
Table 5.2 – Descriptor types and support levels ................................................................. 15
Table 5.3 – Descriptor type and support level ................................................................. 18
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.


[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 040016), 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 040016 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].
### 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.
An AV/C device shall meet these following requirements:

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

<table>
<thead>
<tr>
<th>root_length</th>
<th>crc</th>
</tr>
</thead>
<tbody>
<tr>
<td>03_16</td>
<td>vendor_ID</td>
</tr>
<tr>
<td>81_16 or C1_16</td>
<td>descriptor leaf or directory offset</td>
</tr>
<tr>
<td>17_16</td>
<td>model_ID</td>
</tr>
<tr>
<td>81_16 or C1_16</td>
<td>descriptor leaf or directory offset</td>
</tr>
<tr>
<td>0C_16</td>
<td>node_capabilities</td>
</tr>
<tr>
<td>D1_16</td>
<td>unit directory offset(AV/C protocol)</td>
</tr>
<tr>
<td></td>
<td>additional fields if necessary</td>
</tr>
<tr>
<td></td>
<td>......</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Descriptor Type</th>
<th>Support level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual descriptor in minimal ASCII subset</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Textual descriptors in other character sets</td>
<td>Optional</td>
</tr>
<tr>
<td>Icon descriptor in the YCbCr (48x48) format</td>
<td>Optional (Mandatory, if any icon descriptor is present)</td>
</tr>
<tr>
<td>Icon descriptors in other icon formats</td>
<td>Optional</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Descriptor Type</th>
<th>Support level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual descriptor in minimal ASCII subset</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Textual descriptors in other character sets</td>
<td>Optional</td>
</tr>
<tr>
<td>Modifiable textual descriptors</td>
<td>Optional</td>
</tr>
<tr>
<td>Icon descriptor in the YCbCr (48x48) format</td>
<td>Optional (Mandatory, if any icon descriptor is present)</td>
</tr>
<tr>
<td>Icon descriptors in other icon formats</td>
<td>Optional</td>
</tr>
<tr>
<td>Modifiable icon descriptors</td>
<td>Optional</td>
</tr>
</tbody>
</table>

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.
5.1.4 Node Capabilities entry

The node_capabilities field contains subfields specified by ISO/IEC 13213:1994[3]. 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 0083C016.

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.
5.3.2 Specifier_ID and Version

The specifier_ID shall be 00A02D16 and the version shall be 01000116 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 directory.

Table 5.3 – Descriptor type and support level

<table>
<thead>
<tr>
<th>Descriptor Type</th>
<th>Support level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual descriptor in minimal ASCII subset</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

Unit directory

<table>
<thead>
<tr>
<th>17_{16}</th>
<th>model_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>81_{16}</td>
<td>textual descriptor leaf offset</td>
</tr>
</tbody>
</table>

Textual descriptor leaf

Figure 5.4 – Entries of the Unit directory

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.

![Diagram of Icon descriptor leaf format](image)
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.

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].
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.

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 F000 040016 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 vendor 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.

![Configuration ROM example for a module that has one node](image)

**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 FFFFF₁₆ for the `vendor_ID`, the `model_ID`, and the `eui_64` are for example only. Do not use these values in a real device.

![Configuration ROM example for simple AV/C device](image_url)

---

Copyright © 2001, 1394 Trade Association. All rights reserved.
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 FFFFFF₁₆ for the vendor_ID, the model_ID, and eui_64 are for example only. Do not use these values in a real device.
Figure C.2 – Example of configuration ROM for simple AV/C device with modifiable descriptor

<table>
<thead>
<tr>
<th>bus_info_length</th>
<th>crc_length</th>
<th>crc</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;1&quot;</td>
<td>&quot;3&quot;</td>
<td>&quot;9&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>directory_length</th>
<th>crc</th>
<th>04 04 EA BF</th>
<th>0400</th>
</tr>
</thead>
<tbody>
<tr>
<td>vendor_ID</td>
<td></td>
<td>31 33 39 34</td>
<td>0404</td>
</tr>
<tr>
<td>model_ID</td>
<td></td>
<td>E0 64 61 02</td>
<td>0408</td>
</tr>
<tr>
<td>node_capabilities</td>
<td></td>
<td>0C 00 83 C0</td>
<td>0428</td>
</tr>
<tr>
<td>directory_offset</td>
<td></td>
<td>03 FF FF FF</td>
<td>0418</td>
</tr>
<tr>
<td>model_ID</td>
<td></td>
<td>17 FF FF FF</td>
<td>0420</td>
</tr>
<tr>
<td>vendor_name</td>
<td></td>
<td>81 00 00 0D</td>
<td>041C</td>
</tr>
<tr>
<td>model_name</td>
<td></td>
<td>17 00 00 08</td>
<td>0424</td>
</tr>
<tr>
<td>unit_address_hi</td>
<td></td>
<td>0C 00 83 C0</td>
<td>0428</td>
</tr>
<tr>
<td>unit_address_lo</td>
<td></td>
<td>03 FF FF FF</td>
<td>0418</td>
</tr>
<tr>
<td>max_descriptor_size</td>
<td></td>
<td>04 00 00 0D</td>
<td>044C</td>
</tr>
<tr>
<td>descriptor_address_hi</td>
<td></td>
<td>00 05 09 15</td>
<td>0450</td>
</tr>
<tr>
<td>descriptor_address_lo</td>
<td></td>
<td>00 02 7B A3</td>
<td>0444</td>
</tr>
<tr>
<td>descriptor_address_hi</td>
<td></td>
<td>81 00 00 08</td>
<td>0448</td>
</tr>
<tr>
<td>descriptor_address_lo</td>
<td></td>
<td>00 FF FF FF</td>
<td>0448</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>leaf_length</th>
<th>crc</th>
<th>00 00 00 00</th>
<th>0464</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>character_set</td>
<td>language</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>&quot;M&quot;</td>
<td>&quot;o&quot;</td>
<td>&quot;d&quot;</td>
<td>&quot;e&quot;</td>
</tr>
<tr>
<td>&quot;t&quot;</td>
<td>&quot;n&quot;</td>
<td>&quot;N&quot;</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>&quot;m&quot;</td>
<td>&quot;e&quot;</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>leaf_length</td>
<td>crc</td>
<td>00 05 7F 16</td>
<td>0468</td>
</tr>
<tr>
<td>descriptor_start_for_model_name</td>
<td></td>
<td>00 00 00 00</td>
<td>046C</td>
</tr>
<tr>
<td>width</td>
<td>character_set</td>
<td>language</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>&quot;M&quot;</td>
<td>&quot;o&quot;</td>
<td>&quot;d&quot;</td>
<td>&quot;e&quot;</td>
</tr>
<tr>
<td>&quot;t&quot;</td>
<td>&quot;n&quot;</td>
<td>&quot;N&quot;</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>&quot;m&quot;</td>
<td>&quot;e&quot;</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>max_descriptor_size</td>
<td>descriptor_address_hi</td>
<td>00 06 FF FF</td>
<td>0484</td>
</tr>
<tr>
<td>descriptor_address_lo</td>
<td></td>
<td>00 00 00 00</td>
<td>0488</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>leaf_length</th>
<th>crc</th>
<th>00 05 99 B3</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>character_set</td>
<td>language</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>&quot;M&quot;</td>
<td>&quot;y&quot;</td>
<td>&quot;d&quot;</td>
<td>&quot;M&quot;</td>
</tr>
<tr>
<td>&quot;o&quot;</td>
<td>&quot;c&quot;</td>
<td>&quot;e&quot;</td>
<td>&quot;i&quot;</td>
</tr>
<tr>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
</tbody>
</table>
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 FFFFFF₁₆ for the vendor_ID, the model_ID, and eui_64 are for example only. Do not use these values in a real device.

```
<table>
<thead>
<tr>
<th>unit</th>
<th>Rom Base Address</th>
<th>:hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>:FFFF F000 0400₁₆</td>
<td></td>
</tr>
</tbody>
</table>
```

![Configuration ROM for AV/C device with multiple protocols](image)

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