TA Document 1999028

AV/C Monitor Subunit Model and Command Set Version 1.0

October 24, 2000

Sponsored by:
1394 Trade Association

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

Abstract:
This specification defines a model and command set for the monitor subunit based on the Function Control Protocol defined by IEC 61883. The monitor subunit provides resource information and functionality that are related to monitors. The resource description and control commands for monitor functions are defined.

Keywords:
1394, AV, Protocol, Monitor, Video, MPEG, On Screen Display, OSD.

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

Table of Contents

List of Figures

List of Tables

1. Overview

1.1 Purpose

This document describes the resources and functions of the monitor subunit model as it exists on a 1394 network. The monitor subunit processes video streams and contains methods to control various monitor resources.

The monitor subunit employs the function block mechanism, which provides information about available resources contained in the monitor device and the ability to control these resources. For this purpose, the command set addressed to function blocks is defined, and all of the functions implemented to a monitor subunit are controlled by them.

1.2 Scope

The monitor subunit model and command set described in this specification are intended to be applied to all units that will include a video monitor or capabilities of video processing.

2. References

The following standards contain provisions, which through references in this specification 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] IEC 61883-1,Consumer audio/video equipment - Digital interface - Part 1: General.

[R3] AV/C Digital Interface Command Set General Specification, version 3.0. TA document number 1998003.

[R4] Draft AV/C Panel Subunit Model and Command set version 1.0FC1.

[R5] EIA-775 DTV 1394 Interface Specification.

[R6] AV/C Audio Subunit Specification, version 1.0, TA document number 1999008.

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 set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

3.1.6 reserved values: A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

3.2 Glossary of terms

3.2.1 audio: Digital data or analog signals that carry information for producing sound.

3.2.2 byte: Eight bits of data, used as a synonym for octet.

3.2.3 channel: A specific audio, video or other data stream within a group of streams called clusters.

3.2.4 cluster: A bundle of channels.

3.2.5 content: Streams of video and audio that the subunit decodes and processes. Data in the stream other than video or audio is not considered content.

3.2.6 display device: The physical device that displays still and moving pictures.

3.2.7 function block: A logical unit that has a controllable functionality. For more information see the function block definition in the AV/C Audio Subunit Specification 1.0 [R6].

3.2.8 image: A still picture that is taken from a stream.

3.2.9 video monitor device: A physical device which includes video signal processing and a video display.

3.2.10 OSD: On Screen Display. A secondary overlaid image which is generated by a unit internally or which is generated by an external unit and carried through EIA-775[R5] or a Panel subunit[R4] data connection.

3.2.11 picture: The still or moving images which are generated from a video stream.

3.2.12 resource: Functions and data that are provided by function blocks implemented in a unit or subunit.

3.2.13 signal processing device: A physical device that processes and/or decodes audio, video, and/or other types of data.

3.2.14 speaker device: A physical device which produces sounds from an audio stream.

3.2.15 screen: The location for displaying video and images.

3.2.16 stream: Data transferred between units or internally within a unit which contains digitized signals (in case encoded data), or analog signals.

3.2.17 tuner device: A physical device which tunes to one or several broadcast signals and channels and processes them.

3.2.18 TV/DTV: A television/digital television, which has the capability of receiving and processing broadcast signals. Functions of TV/DTV are usually performed by a display device, one set of speaker devices, necessary signal processing devices, and one or more tuner devices.

3.2.19 video: A data stream or analog signals that carry picture information.

3.3 Acronyms and abbreviations

AV/C	Audio Video Control
AV	Audio and video
FB	Function Block

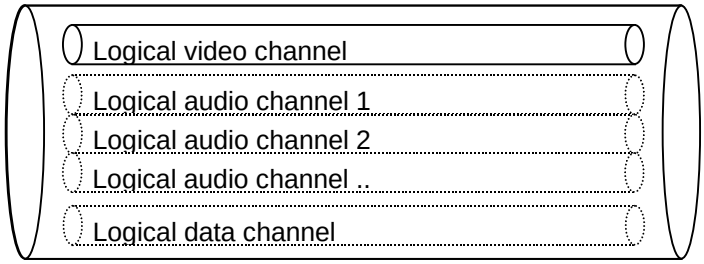
4. Monitor subunit model

The monitor subunit is a logical entity in a monitor device that allows other nodes on a 1394 network to recognize, access the available resources of, and use functions of that device. All resources and functions of a monitor device are accessed from the monitor subunit.

The monitor subunit comprises function blocks for video processing. In this specification, these function blocks, their functionality, and their basic connections are defined. The monitor subunit employs the function block structure that is defined in the AV/C Audio Subunit Specification 1.0 [R6].

4.1 AV channel clusters

The basic data structure of a stream that is shown in the monitor subunit is called AV channel cluster. The following figure illustrates a conceptual model of the AV channel cluster.



The solid line channel is used by the monitor subunit, the dashed line channels are not used.

Figure 4.1 – AV channel cluster

The AV channel cluster in a monitor subunit has one logical video channel as defined by this specification.

— Also, the AV channel cluster is designed to be able to support other channels for future extension like multiple video channels or data channels.

The monitor subunit uses two kinds of channel clusters: a master AV channel cluster and subset AV channel clusters used by function blocks. The master AV channel cluster is a superset of all the other AV channel clusters, and contains all the possible channels that can be supported by the subunit. For more information about the master AV channel cluster and the other AV channel clusters, refer to section 5.3 “Cluster Information.”

In this specification, all logical channels in an AV channel cluster are format independent.

The channels in the master AV channel cluster and the other AV channel clusters are described in the monitor subunit identifier descriptor. Since a channel cluster with multiple video channels or any audio and data channels are not supported in this version, only one channel configuration type is defined for a cluster that contains a single video channel. The monitor subunit’s function blocks have no effect on audio or data channels, and if those types of channels are in the input stream, the monitor subunit passes through those channels.

4.2 Components of the monitor subunit model

4.2.1 Destination plugs

A monitor subunit has one or more subunit destination plugs. Streams that enter these plugs may have video, audio, and/or other types of content. An input stream on a destination plug may be decoded by, filtered through, duplicated by and/or simply passed through the destination plug, depending on the format of the input stream and how the destination plug's output is routed to function blocks. However, the output of the destination plug shall constitute the master AV channel cluster. The master AV channel cluster specifies the streams that the monitor subunit can process. The destination plug shall not strip and route individual channels from the AV channel cluster, but shall provide all destination function blocks with the same master AV channel cluster.

4.2.2 Source plugs

A monitor subunit may also have subunit source plugs that pass through data it receives from its subunit destination plugs. The monitor subunit may pass through video content with or without modifications and passes through audio and/or other types of content without any modifications.

4.2.3 Function blocks

The monitor subunit uses the function block mechanism to control video streams. Each function block has one or more input plug(s) and zero or one output plug. A function block receives data from either a subunit destination plug or another function block through an input plug, and it captures and processes all or part of the channels represented in the AV cluster for that function block. Each function block has unique AV channel cluster for its output plug, which defines a subset of the channels in the master AV cluster. The AV channel cluster that enters the input plug depends on its upstream plug or function block. Any channel that a function block does not process is either passed through the function block or is terminated within the function block. In this specification, two types of function block are defined. The video feature function block is defined to control the characteristics of video, and the display function block is defined to display the video channel.

New function blocks can be added for more advanced features in a future revision of this specification.

4.3 Monitor subunit connections

This specification defines a particular configuration of destination plugs, function blocks, and their connections for a monitor subunit with basic functionality. Monitor subunits can include different functionality by adding or removing video function blocks and changing connections as desired by the implementation.

The following figure shows an example of a Monitor Subunit configuration.

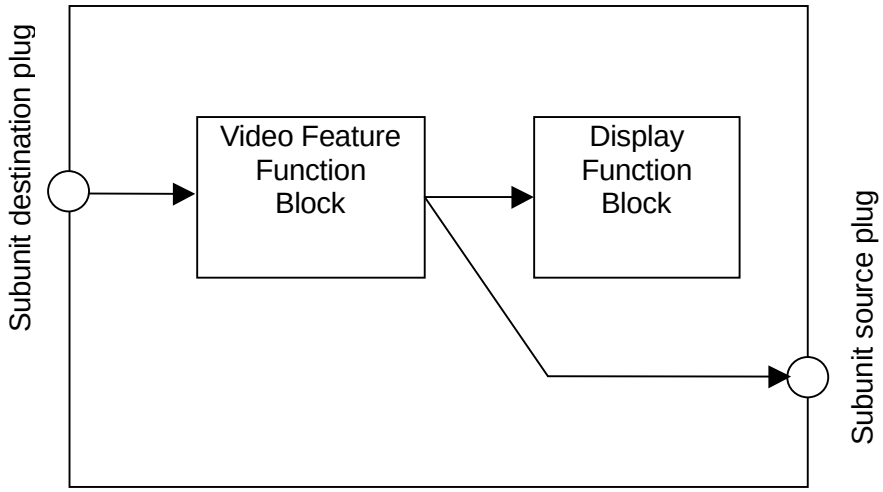


Figure 4.2 – Monitor subunit example

This example shows a monitor subunit with a video feature function block and a display function block (both defined in this specification). A stream in the form of the master AV channel cluster is delivered to the video feature function block, which is used to process the video stream and is serially delivered to a display function block, which is used to display the video stream on a display device. Furthermore, the output stream from the video feature function block is connected to the subunit source plug.

4.4 Function blocks in the monitor subunit

The function block mechanism is designed so that new functions can be easily implemented. For details of the function block mechanism, refer to the AV/C Audio Subunit Specification 1.0 section 5.5, “Function Blocks and Function Block Plugs” [R6].

In this specification, a video feature function block is defined to control the characteristics of video, and a display function block is defined to terminate and display a video stream.

Although only two function blocks are defined in this specification, new function blocks can be added for more advanced functions in a future revision of this specification.

4.4.1 Video feature function block

The video feature function block can be used to change the characteristics of the video; such as brightness, contrast, and color. Details of the changeable characteristics are described in section 6.1, “Video feature function block control command”. The video feature function block has one input plug and one output plug as shown in the figure below.

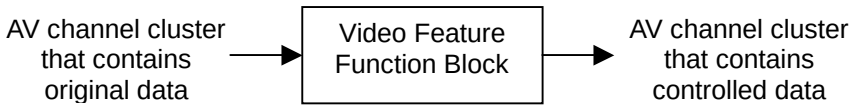


Figure 4.3 – Video feature function block for controlling video

4.4.2 Display function block

The display function block terminates the video stream, and is responsible for displaying video content on the monitor device. The display function block has one or more input plugs and has no output plug as shown in the figure below.

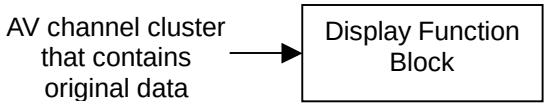


Figure 4.4 – Display function block for video termination

If a unit has a display device such as a CRT, its monitor subunit should have a display function block and, the display function block should have at least one input plug. If a display function block has several input plugs, the display device can display several overlaid images on the screen.

4.5 Configurations

The monitor subunit is formed by the function blocks described above, and their configurations are provided in the *configuration_dependent_information* field in the subunit identifier descriptor (Refer to section 5.2, “Monitor subunit identifier descriptor”). This information describes the function blocks that exist within the subunit, as well as the plugs, master cluster information, and output cluster information of these function blocks, and also the way their connections configured. A monitor may have one or more configurations. Controllers learn these configurations by using descriptor commands or by using available function block control and status commands.

4.6 Information access model

The monitor subunit, particularly the video feature function block, provides information about the capabilities and status of monitor functions in an AV unit. Other devices can use this information to learn about and control the monitor subunit.

4.6.1 Static information access

Controllers may require static information such as the capabilities of the monitor subunit. It is provided by a descriptor (described in section 5, “Descriptor information”) and can be obtained using the OPEN DESCRIPTOR and READ DESCRIPTOR commands. See the AV/C General Specification [R3].

4.6.2 Dynamic information access

Dynamic information such as the changeable characteristics of the monitor can be retrieved using function block commands with a ctype of STATUS. Please refer to the AV/C Audio Subunit Specification 1.0 section 10, “Function Block Command” [R6] for general issues of function block control.

4.7 OSD data connection

In the monitor subunit model, one or more subunit destination plugs are provided for input of OSD data, which is transferred asynchronously. It is necessary for the monitor subunit to connect the destination plug(s) to the serial bus asynchronous input plug of its AV unit. OSD data shall be decoded into an OSD image at the subunit destination plug and transferred into the subunit as a logical video channel in an AV channel cluster. Control of OSD data is defined by their related specifications (EIA-775[R5], AV/C Panel Subunit [R4]).

NOTE — Since there is no distinction between subunit destination plugs for AV streams and for OSD data, the monitor subunit has to manage plug assignment based on the data format received. The plug assignment means that a target may have to decide the destination plug number of the monitor subunit to allocate when it receives a CONNECT control command specifying “any destination plug”.

5. Descriptor information

5.1 Descriptor structure

The monitor subunit contains a subunit identifier descriptor and its structure is shown in the following figure. The following pages describe the subunit identifier descriptor fields in detail. The structure definition of the *subunit_dependent_information* is modeled after AV/C Audio Subunit Specification 1.0 section 6, "Audio Subunit Identifier Descriptor" [R6].

Subunit Identifier Descriptor
descriptor_length
generation_ID
size_of_list_ID
size_of_object_ID
size_of_object_position
number_of_root_object_lists = n
root_object_list_id 0
:
root_object_list_id n-1
monitor_subunit_dependent_length
monitor_subunit_dependent_information
configuration_dependent_information
master_cluster_information
function_block_dependent_information 1
cluster_information
function_block_type_dependent_information
:
function_block_dependent_information m
manufacturer_dependent_length[1]
manufacturer_dependent_information [1]

[1] *manufacturer_dependent_information* is not defined in this document.

Figure 5.1 – Structure of Subunit Identifier Descriptor for the Monitor Subunit

5.2 Monitor subunit identifier descriptor

The monitor subunit identifier descriptor is modeled after the subunit identifier descriptor defined in reference [R3].

The *generation_ID* field for the monitor subunit shall be 01_{16}.

The monitor subunit has no object lists. The following configurations are applied to the subunit identifier descriptor for the monitor subunit version 1.0.

The *size_of_list_ID* field for the monitor subunit shall be 00_{16}.

The *size_of_object_ID* field for the monitor subunit shall be 00_{16} (no *object_IDs* necessary).

The *size_of_object_position* field for the monitor subunit shall be 00_{16}.

The *number_of_root_object_lists* field for the monitor subunit shall be 00_{16}.

The following table shows the subunit specific information fields for the monitor subunit.

Address Offset	Length, bytes	External Read/Write	Static/ Dynamic	Contents
$00\ 00_{16}$ $00\ 01_{16}$	2	R	S	monitor_subunit_dependent_info_fields_length = L
$00\ 02_{16}$	1	R	S	monitor_subunit_version
$00\ 03_{16}$	1	R	S	number_of_configurations (01_{16})
$00\ 04_{16}$: 0004_{16}+L-1	L - 1	R	S	configuration_dependent_information…

Figure 5.2 – Monitor subunit dependent information

The *monitor_subunit_dependent_info_fields_length* field contains the length in bytes, which begins from *monitor_subunit_version* field and continues to the end of the *configuration_dependent_information* field.

The *monitor_subunit_dependent_info_fields_length* is introduced so that extended fields can follow the existing fields, in case the monitor subunit dependent information needs to be expanded in the future. Controllers can easily determine if any extended fields exist here by comparing the *monitor_subunit_dependent_length* and *monitor_subunit_dependent_info_fields_length* fields. In AV/C Monitor Subunit Specification Version 1.0, the value of the *monitor_subunit_dependent_info_fields_length* field equals to the *monitor_subunit_dependent_length* minus two. If the following formula is true:

monitor_subunit_dependent_length > (*monitor_subunit_dependent_info_fields_length* + 2)

then extended fields exist in this structure.

The *monitor_subunit_version* field contains the version number of the monitor subunit. This field shall be set to 00_{16} for version 1.0.

The *number_of_configurations* field contains the number of *configuration_dependent_information* fields. In the monitor subunit specification version 1.0, this value shall be 01_{16}.

Note that all of the data structures are static (S) and read-only (R). That is, the size of the data in the monitor subunit identifier descriptor shall not change at any time.

Address Offset	Length, bytes	External Read/Write	Static/ Dynamic	Contents
$00\ 00_{16}$	2	R	S	configuration_dependent_information_length = L
$00\ 01_{16}$				
$00\ 02_{16}$	2	R	S	configuration_ID
$00\ 03_{16}$				
$00\ 04_{16}$	m	R	S	master_cluster_information…
:				
$00\ 04_{16}$+m	1	R	S	number_of_subunit_source_plug_link_information = n
$00\ 05_{16}$+m	2	R	S	subunit_source_plug[0]_link_information…
:				
:				:
0005_{16}+m+2*(n-1)	2	R	S	subunit_source_plug[n-1]_link_information…
:				
0005_{16}+m+2*n	1	R	S	number_of_function_block_dependent_information = u
0006_{16}+m+2*n	-	R	S	function_block_dependent_information[1]…
:	:	R	S	:
$(L - 2)_{16}$	-	R	S	function_block_dependent_information[u]…

Figure 5.3 – Configuration dependent information

The *configuration_dependent_information_length* field contains the length in bytes, which begins from the *configuration_ID* field and continues to the end of *function_block_dependent_information* fields.

The *configuration_ID* field contains the ID number which is unique to the subunit configuration. The ID may be used to identify the configuration as the preset number of configuration.

The *master_cluster_information* field contains the cluster information of the master cluster. Refer to Figure 5.6 for its contents.

The *number_of_subunit_source_plug_link_information* field contains the number of the following *subunit_source_plug[x]_link_information* fields.

The *subunit_source_plug[x]_link_information* fields contain the function block type and function block ID or subunit destination plug number that is connected to the subunit source plug number x. The detail data structure of each *subunit_source_plug[]_link_information* field is described in Figure 5.4. See Table 6.3 for function block type encoding and refer to Figure 5.5 for function block ID information.

offset	Contents		
	Source is fb-plug	Source is subunit destination plug	Not connected
$00\ 00_{16}$	function block type	$F0_{16}$ (subunit destination plug)	FE_{16} (Not connected)
$00\ 01_{16}$	function block ID	subunit destination plug number	XX_{16}(ignored)

Figure 5.4 – Subunit_source_plug[]_link_information and Source_ID[] fields

The *number_of_function_block_dependent_information* field contains the number of the *function_block_dependent_information* fields, which is equal to the number of function blocks in the monitor subunit.

The *function_block_dependent_information* is shown in the following figure.

Address Offset	Length, bytes	External Read/Write	Static/ Dynamic	Contents
00 00_{16} 00 01_{16}	2	R	S	function_block_dependent_information_length = L
00 02_{16}	1	R	S	function_block_type
00 03_{16}	1	R	S	function_block_ID
00 04_{16} 00 05_{16}	2	R	S	reserved (00 00_{16})
00 06_{16}	1	R	S	number_of_input_plugs[1] = m
00 07_{16} :	2	R	S	source_ID [0] [1]…
:				:
00 07_{16} + 2*(m-1) :	2	R	S	source_ID [m -1] [1]…
00 07_{16} + 2*m	n	R	S	cluster_information...
00 07_{16} + 2*m+n	2	R	S	function_block_type_dependent_information_length = p
00 09_{16} + 2*m+n : (L - 2)$_{16}$	p	R	S	function_block_type_dependent_information...

[1] These definitions are same as those in AV/C Audio Subunit Specification 1.0 section 9.1 Common Function Block Dependent Information[R6].

Figure 5.5 – Function block dependent information

The *function_block_dependent_information_length* field contains the length in bytes, which begins from *function_block_type* field and continues to the end of *function_block_type_dependent_information* field.

The *function_block_type* field contains the function block type. Refer to function block type encoding in Section 6.2.

The *function_block_ID* field contains the function block ID, which is unique to each function block type in a subunit

The *number_of_input_plugs* field contains the number of input plugs in the function block.

The *source_ID[m]* field contains a combination of the function block type and the function block ID if the input function block plug[m] is connected to the output function block plug of other function block. Or it contains a combination of the value $F0_{16}$(subunit destination plug) and the subunit destination plug number connected to the input function block plug[m] if the input plug is connected to the subunit destination plug. If the input function block plug[m] is not connected, the value for function block type shall be set FE_{16} and the value for function block ID shall be ignored. The Figure 5.4 describes the values in the *Source_ID[]* field. The *cluster_information* field contains the cluster information of the output function block plug, which is described by the cluster information data structure. For details of the cluster information, refer to Figure 5.7.

For *function_block_type_dependent_information_length* and *function_block_type_dependent_information* fields are defined in the table below.

Table 5.1 – Function Block Type Dependent Information Length and Function Block Type Dependent Information fields

Function block type	*function_block_type_dependent_information_length* field	*function_block_type_dependent_information* field
Display function block	00_{16}	None.
Video feature function block	00_{16}	None.

5.3 Cluster information

This section describes the detail of information about the master cluster and the cluster handled in a function block. Figure 5.6 shows the structure of master cluster.

Address Offset	Length, bytes	External Read/Write	Static/ Dynamic	Contents
00 00_{16}	2	R	S	master_cluster_information_length = L
00 01_{16}				
00 02_{16}	2	R	S	video_cluster_length = m
00 03_{16}				
00 04_{16}	1	R	S	number_of_video_channels
00 05_{16}	1	R	S	video_channel_configuration_type

Figure 5.6 – Master cluster information in configuration dependent information

The *master_cluster_information_length* field contains the length in bytes between *video_cluster_length* field and *video_channel_configuration_type* field at the end of the master cluster information fields.

The *master_cluster_information_length* and *video_cluster_length* fields are introduced so that extended fields can follow the existing fields, in case the master cluster information needs to be expanded in the future. Controllers can easily determine if any extended fields exist here by comparing the *master_cluster_information_length* and *video_cluster_length* fields. In AV/C Monitor Subunit Specification Version 1.0, the value of the *master_cluster_information_length* field equals to the *video_cluster_length* minus two. If the following formula is true, it indicates that the target has some extended fields conforming to a future extension:

master_cluster_information_length > (*video_cluster_length* + 2)

The *video_cluster_length* field shows the length of the fields used for information about the video cluster, which begins from the *number_of_video_channels* field and continues to the *video_channel_configuration_type* field. Since only SINGLE_VIDEO is defined for *video_channel_configuration_type* in this version, the value of *video_cluster_length* field is 02_{16}.

The *number_of_video_channels* shall be set to 01_{16} in version 1.0 of this specification.

The *video_channel_configuration_type* field contains the ID for the video channel structure, which is encoded as 10_{16} for SINGLE_VIDEO in version 1.0. Refer to Table 5.2.

The cluster information in function block dependent information is given in the following figure:

Address Offset	Length, bytes	External Read/Write	Static/ Dynamic	Contents
00 00_{16}	2	R	S	cluster_information_length = L
00 01_{16}				
00 02_{16}	2	R	S	video_cluster_length = m
00 03_{16}				
00 04_{16}	1	R	S	number_of_video_channels
00 05_{16}	1	R	S	video_channel_configuration_type

Figure 5.7 – Cluster information in function block dependent information

The *cluster_information_length* field contains the length in bytes between the *video_cluster_length* field and the *video_channel_configuration_type* field at the end of the cluster information.

The *cluster_information_length* and *video_cluster_length* fields are introduced so that extended fields can follow the existing fields, in case the master cluster information needs to be expanded in the future. Controllers can easily determine if any extended fields exist here by comparing the *cluster_information_length* and *video_cluster_length* fields. In AV/C Monitor Subunit Specification Version 1.0, the value of the *cluster_information_length* field equals to the *video_cluster_length* minus two. If the following formula is true, it indicates that the target has some extended fields conforming to a future extension:

cluster_information_length > (*video_cluster _length* + 2)

The *video_cluster_length* field shows the length of the fields used for information on the video cluster, which begins from *number_of_video_channels* field and continues to the end of *video_channel_configuration_type* field. Since only SINGLE_VIDEO is defined for *video_channel_configuration_type* in this specification, the value of *video_cluster_length* field is 02_{16}.

The *number_of_video_channels* shall be set to 01_{16} in version 1.0 of this specification.

The *video_channel_configuration_type* field contains the ID for video channel structure, which is encoded value 10_{16} for SINGLE_VIDEO in version 1.0. Refer to Table 5.2.

Table 5.2 – video_channel_configuration_type encoding

Value	Meaning
00_{16}	Same as master cluster [1]
01_{16}	Same as upstream [1]
02_{16} - $0F_{16}$	Reserved
10_{16}	SINGLE_VIDEO
11_{16} - FF_{16}	Reserved

[1] These definitions are same as those in AV/C Audio Subunit Specification 1.0 section 6.1.2 Cluster information [R6].

6. Subunit commands

All functions of a monitor subunit are controlled by function block commands. For more information on function block commands, refer to the AV/C Audio Subunit Specification 1.0 section 10, Function Block Command [R6].

The following figure shows the format of the function block command used to control the monitor subunit.

	msb							lsb
	00 00					ctype		
	subunit_type (monitor subunit = 00_{16})					subunit_ID		
opcode	FUNCTION_BLOCK command ($B8_{16}$)							
operand[0]	function_block_type							
operand[1]	function_block_ID							
operand[2]	control_attribute							
operand[3]	selector_length = n + 1							
operand[4]	video_selector_data[1]							
:	:							
operand[3+n]	video_selector_data[n]							
operand [3+n+1]	control_selector							
operand [4+n+1]	control data							

Figure 6.1 – Format of the function block command

The fields shown in non-shadow are defined in the audio subunit document [R6], and in the AV/C General document [R3].

The value of the *selector_length* field includes the *video_selector_data* and the *control_selector*. It does not include itself.

The *video_selector_data[x]* fields contain information for video channel selection such as input FB-plug number, input or output *video_channel_number*. The detail of this field is specified in each function block type.

The *control_selector* field indicates which type of control this command is manipulating.

The control data field has *control_selector* specific data. The detail of this field is described in the following clauses.

6.1 Video feature function block control command

The function parameters defined for the video feature function block are shown in percentages (%). Valid values are, therefore, integer values equal to or between -100 and +100.

The video feature function block may apply the following function block control attributes.

— CURRENT:
To get the setting of a control, use this attribute with a ctype of STATUS or NOTIFY. To define the current setting for a control, use this attribute with a ctype of CONTROL.

— MAXIMUM & MINIMUM:
Represents the value of the upper boundary and the lower boundary for the control respectively with ctype of STATUS.

— RESOLUTION:
Represents a minimum scale resolution of the control with a ctype of STATUS and is always positive. The value of resolution can take a number in the range of 1 to 100.

— DEFAULT:
Defines the default setting for the control with a ctype of CONTROL. To get the setting of a control, use this attribute with a ctype of STATUS or NOTIFY.

— DELTA:
Defines the relative change of a control. This control attribute is used only with a ctype of CONTROL. The value of the control attribute is not specified directly. Instead, a multiple of the value in RESOLUTION shall be specified.

The following figure shows the function block type dependent data of the function block command for video feature function block.

	msb							lsb
operand[3]	02_{16}							
operand[4]	video_channel_number (IVCN)							
operand[5]	control_selector							
operand[6]	Parameter							

Figure 6.2 – Function block type dependent data for video feature function block controls

Operand[3] is the number of video selector data in bytes. For the video feature function block, it shall be 02_{16}, which contains one byte for each *video_channel_number* (IVCN: Input Video Channel Number) and *control_selector* fields.

The *video_channel_number* (IVCN) indicates the video channel number of the AV channel cluster to be processed. In this version, it shall be 00_{16}(master control), 01_{16} (specifies a channel) or FF_{16} (specifies all channels in the cluster information of each function block). This is because there is only one video channel in SINGLE_VIDEO video channel configurtation type of an AV channel cluster, and the following fields have the same values in all cases.

Applicable values for the *control_selector* field are shown in the table below.

Table 6.1 – *control_selector* encoding

Value	Name	Meaning
01_{16}	BRIGHTNESS	Control brightness of the video
02_{16}	CONTRAST	Control contrast of the video
03_{16}	COLOR	Control color of the video
other values	-	Reserved

Applicable values for the *parameter* field for each *control_selector* is shown in section 6.1.1 “Value representation”. The values actually implemented to a target varies and can be obtained by using control attributes of MAXIMUM, MINIMUM and RESOLUTION with ctype of STATUS.

6.1.1 Value representation

The values of parameters of the command frame are equal or between -100 and +100, shown in two’s complement. All other values are reserved.

Table 6.2 – Parameter encoding

Value	Meaning
00_{16} – 64_{16} [1]	0_{10} – 100_{10} [1]
$9C_{16}$ – FF_{16} [1]	-100_{10} – -1_{10} [1]
$7F_{16}$	Invalid
other values	Reserved

[1] two's complement value

6.2 Display Function Block control command

As of version 1.0 of this specification, the display function block has no associated commands.

6.3 Function block type encoding

Table 6.3 – Function_block_type encoding

Function Block Type Identifier	Meaning
00_{16} - $7F_{16}$	Reserved for general use[1].
80_{16} - $8F_{16}$	Reserved (used in Audio Subunit Specification[1])
90_{16} - $9F_{16}$	Reserved
$A0_{16}$	Display function block
$A1_{16}$	Video Feature function block
$A2_{16}$ - EF_{16}	Reserved
$F0_{16}$	Subunit destination plug
$F1_{16}$	Subunit source plug
$F2_{16}$ - FD_{16}	Reserved
FE_{16}	Not connected
FF_{16}	Reserved

[1] This is defined in the AV/C Audio Subunit Specification 1.0 section 10.1.2 Function_block_type and Table 10-1 – Function block type encoding[R6]

This page is left intentionally blank

Annex

Annex A: Monitor subunit identifier descriptor overview

The following figure is an overview of the monitor subunit identifier descriptor.