TA Document 1999031

AV/C Connection and Compatibility Management Specification 1.0

July 10, 2000

Sponsored by:
1394 Trade Association

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

Abstract:
The Connection and Compatibility Management (CCM) specification describes procedures and commands for setup and maintenance of connections that realize data transmission between AV devices on a 1394 bus.

Keywords:
Connection, Compatibility, Management.

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

List of Figures

List of Tables

1. Overview

This document specifies a mechanism for Connection and Compatibility Management (CCM) between AV/C Devices on a 1394 network, and a command set to be used for it. This document describes procedures and commands for setup and maintenance of connections.

The purpose and scope of this specification are summarized below.

1.1 Purpose

The purpose of this specification is to provide a mechanism that enables the following:

- The ability of a controller to set up the internal source and destination connections between subunit and unit plugs of a device, regardless of the device's internal connection configuration.
- The ability of a controller to instruct a destination device to set up an end-to-end connection with a source device.
- The ability of a destination device to respond to changes in the isochronous channel of a source device.
- The ability of a device to establish preset default connections to another device.

1.2 Scope

This specification builds on the information provided by IEEE Std 1394-1995, Standard for a High Performance Serial Bus [R1] and IEC 61883-1, Consumer audio/video equipment – Digital interface – Part 1: General [R2], and provides methods for making internal connections in a unit, and end-to-end connections between units.

This specification contains scenarios of how connections are set up by a user, and defines the underlying procedures and AV/C commands that realize the scenarios. The commands defined in this document conform the AV/C Digital Interface Command Set General Specification[R3], and the procedures include connections defined in IEC 61883-1 [R2].

Considered connection types are summarized below:

Peripheral connections between devices:

- Isochronous serial bus connections
- Asynchronous serial bus connections (specified by a future version of this document.)
- External connections

Internal connection within a device:

- Subunit destination and source plugs.
- Subunit destination plugs and unit input plugs.
- Subunit source plugs and unit output plugs.
- Unit input and output plugs.

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] 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] Enhancements to the AV/C General Specification 3.0, Version 1.0, January 26,1999

[R5] AV/C Tuner Model and Command Set Version 1.0, April 15,1998

[R6] Audio Subunit Specification Version 1.0, TA document number 1998003

[R7] AV/C Compatible Asynchronous Serial Bus Connections, Version 1.0, January 26, 1999

[R8] AV/C commands for management of Asynchronous Serial Bus Connections, Version 1.0, January 26, 1999

[R9] AV/C Tape Recorder/Player Subunit Specification, Version 2.1, January 26, 1999

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.

NOTE — The IEEE is investigating whether the "may, shall, should" and possibly "expected" terms will be formally defined by IEEE. If and when this occurs, draft editors should obtain their conformance definitions from the latest IEEE style document.

3.2 Glossary of Terms

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

3.2.2 controller: A device that issues AV/C commands to targets.

3.2.3 target: A device that receives AV/C commands from controllers.

3.2.4 source: A subunit source plug, unit output plug, or unit input plug that serves as the originating source of data streams. There are some restrictions on which type of plug can be a source, depending on what type of connection is referred.

3.2.5 destination: A subunit destination plug, unit input plug, or unit output plug that receives data streams transmitted by a source. There are some restrictions on which type of plug can be a destination, depending on what type of connection is referred.

3.2.6 point-to-point connection: Defined by IEC 61883-1, a point-to-point connection is a type of agreement between two devices such that the transmission of an isochronous stream from one device to the other shall be terminated only by the initiating device.

3.2.7 broadcast connection: Defined by IEC 61883-1, a broadcast connection is where one device transmits on an isochronous channel and where other devices "listen" to that channel.

3.2.8 broadcast channel base: Defined by IEC 61883-1, an isochronous channel number given to a device and the device determines a broadcast channel by the number when it establishes a broadcast-out connection on an oPCR.

3.2.9 internal connection: A connection made within a unit.

3.2.10 peripheral connection: A connection made between units.

3.2.11 end-to-end connection: A connection between the originating source and the ultimate destination. An end-to-end connection may include internal connections and peripheral connections.

3.2.12 IRM: isochronous resource manager

3.2.13 isochronous: A term that indicates the essential characteristic of a time-scale or signal, such that the time intervals between consecutive instances either have the same duration or durations that are integral multiples of the shortest duration. In the context of serial bus, "isochronous" is taken to mean a bounded worst-case latency for the transmission of data; physical and logical constraints that introduce jitter preclude the exact definition of "isochronous."

3.2.14 node: An addressable device attached to serial bus with at least the minimum set of control registers defined by IEEE Std 1394–1995.

3.2.15 node_ID: A 16-bit number, unique within the context of an interconnected group of serial buses. The node_ID is used to identify both the source and destination of serial bus asynchronous data packets. It can identify one single device within the addressable group of serial buses (unicast), or it can identify all devices (broadcast). The ten most significant bits of node_ID are the same for all nodes on the same bus; this is the bus_ID. The six least significant bits of node_ID are unique for each node on the same bus; this is called phy_ID.

3.2.16 PCR: Plug Control Register, as defined by IEC 61883, Digital Interface for Consumer Electronic Audio/Video Equipment.

3.2.17 iPCR: Input plug PCR, as defined by IEC 61883.

3.2.18 oPCR: Output plug PCR, as defined by IEC 61883.

3.2.19 plug: A physical or virtual end-point of connection implemented by an AV unit or subunit that may receive or transmit isochronous or other data. Plugs may be serial bus plugs, accessible through the PCR's; they may be external, physical plugs on the AV unit; or they may be internal virtual plugs implemented by the AV subunits.

3.2.20 serial bus: The physical interconnects and higher level protocols for the peer-to-peer transport of serial data, as defined by IEEE Std 1394–1995.

3.2.21 stream: A time-ordered set of digital data originating from one source and terminating at zero or more sinks. A stream is characterized by bounded bandwidth requirements and by synchronization points, or time stamps, within the stream data.

3.2.22 unit architecture: The formal specification of the format and function of the software-visible resources and behaviors of a class of units. This document, in conjunction with the references above, defines a unit architecture for the class of AV devices.

3.3 Acronyms and Abbreviations

AV/C: Audio/video control

IEEE: The Institute of Electrical and Electronics Engineers, Inc.

3.4 Unimplemented locations

3.4.1 Reserved locations

The reserved field shall be set as defined in the section 3.1, "Rules for the reserved fields", page 5 of "Enhancements to the AV/C General Specification 3.0, Version 1.0, January 26,1999"[R4].

4. Connection Management Model

4.1 Physical and logical connections

Serial bus connections differ from traditional analog connections in that when a serial bus physical connection is made, it does not mean that a logical connection is made between the devices. For example, if device A is physically connected to device B in a bus, it doesn't necessarily mean that device A can communicate with device B, but that it now can make logical connections to all nodes on the bus. The following figure illustrates this concept.

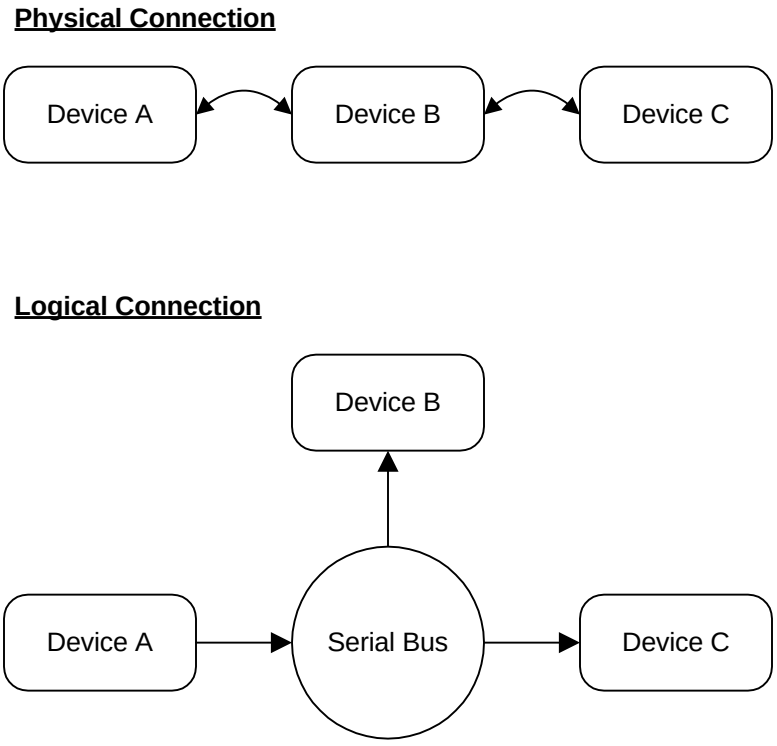


Figure 4.1 – serial bus physical vs. logical model

It is also possible to combine connections that are on a serial bus and those that are on another external interface. This is possible, for example, when connecting a non-serial bus device to an existing serial bus network. In this case, the node that it connects to is the "gateway" to that device on the bus. Using A/D Converters or some other means, a device that is foreign to the serial bus can still communicate with devices on the network via the intermediary node. The following figure shows this example.

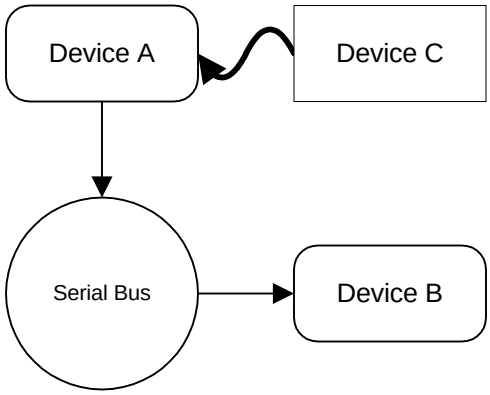


Figure 4.2 – Connection external devices to serial bus

In the figure above, Device C is able to connect to the serial bus network and send data to Device B via Device A's external plug.

4.2 Connection model

4.2.1 Peripheral connections

Peripheral connections consist of isochronous and asynchronous serial bus connections, and external connections.

4.2.1.1 Isochronous connections

Two types of isochronous connections are defined in IEC 61883-1, namely, point-to-point connection and broadcast connection. In this section, these two types of connection and additional concept of monitor connection are introduced.

4.2.1.1.1 Point-to-point connections

IEC 61883-1 describes point-to-point isochronous connections in detail, but leaves out the purpose and practical applications of making point-to-point connections. In a practical point-to-point connection, three things are accomplished:

1) An available iPCR and oPCR are determined. oPCR availability is determined by whether or not the source either has a free oPCR that can transmit the desired content, or already contains a plug that is transmitting the desired content. iPCR availability is determined by whether a destination contains a plug that is presently not receiving but can receive the desired content.

2) For a new connection, an isochronous channel and bandwidth are obtained from the IRM, and the channel and other fields values, are set in the iPCR and oPCR registers according to IEC 61883-1. For an overlaid connection, the oPCR and iPCR are adjusted but the channel is not changed and no extra bandwidth is allocated.

3) Implicitly, an agreement is made that only the initiator of the connection shall be able to disconnect it, which is the reverse of 2) above.

The above points imply that the point-to-point connection is simply an agreement between two units to establish a communications session, and that only one of the units shall control that agreement. Point-to-point connections are important when a device requires uninterrupted communications services from another device, such as in the case of a recording session.

4.2.1.1.2 Broadcast connections

IEC 61883-1 also explains broadcast connections in detail, and also leaves out the practical applications of these kinds of connections. There are two types of broadcast connections; broadcast-out, and broadcast-in. Typically, legacy devices use the default broadcasting channel for all devices on the bus. It is not recommended for a bus to have various broadcast channel bases due to legacy device compatibility issues.

When a device begins to broadcast-out a signal, two things occur:

1) Bandwidth and channel are allocated from the IRM
2) The channel and other values are written to the oPCR according to IEC 61883-1, and transmission begins.

When a device begins a broadcast-in connection, the device sets an available iPCR to "listen-in" to a channel.

No agreement may be made between any devices in this kind of broadcast connection. The connection is only between a device and the bus. Thus, the source device may stop transmitting at any time, or the destination device may stop listening at any time without communicating the disconnection to any other device. Any other device may take over transmitting on the default broadcast channel at any time. Under this condition, all receiving device will begin receiving the new transmission. Furthermore, any device may set up a point-to-point connection that usurps the channel until it finishes with it.

4.2.1.1.3 Monitor connections and virtual output

The monitor connection is a particular type of broadcast connection with an agreement that two devices will listen to the same isochronous channel supplied by a source device. A monitoring device communicates with the initial listening device, and requests to be notified of any isochronous channel changes. Once notified, the monitoring device can then change the channel to which it is listening to match the initial listening device.

To achieve this functionality, the monitoring device connects to a "virtual output" of the initial listening device, and receives information about the channel that is presently being listened to.

The example of "virtual output" is described the figure below. A STB outputs a signal to isochronous channel 0, and a VCR unit inputs the signal for recording through the iPCR. In this case, even if the TV is trying to monitor the signal that the VCR is recording (similar to the analogue system), the VCR may not output the data from its oPCR, or the VCR may output only empty packets from its oPCR. In addition to that, if the VCR actually output the recording signal from oPCR through channel 1, it causes wasted use of the channel and bandwidth. Even though the output monitoring signal is the same as the input recording signal, an additional isochronous channel and bandwidth would be required for the monitoring signal. Thus, instead of the "real output", the TV determines the channel that is input by the VCR, and inputs that channel itself. The output status of the oPCR in the virtual source is "virtual output to channel 0".

The virtual output should be received through a broadcast connection, because the listening device may switch the source device.

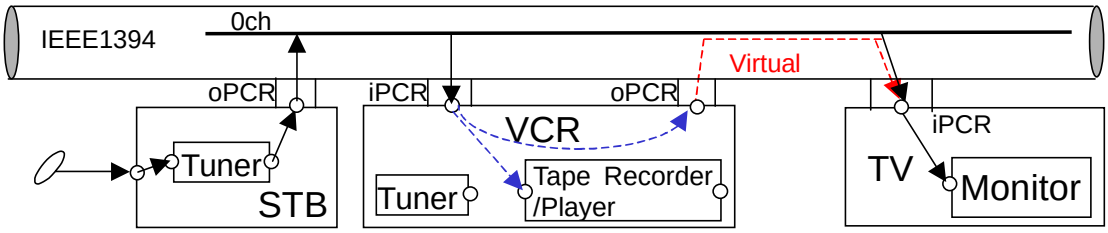


Figure 4.3 – Example of virtual output

4.2.1.2 Asynchronous connections

Asynchronous connections are out of scope of this specification.

4.2.1.3 External connections

External connections are connections that do not include the 1394 serial bus, and can be analog or digital.

4.2.2 Internal connections

Unit internal connections are connections that are established between:

- Subunit destination and source plugs.
- Subunit destination plugs and unit input plugs.
- Subunit source plugs and unit output plugs.
- Unit input and output plugs.

The configuration of unit internal connections is implementation dependent. Some connections may be hard-wired, and some may switchable. Internal connections are set up and managed by AV/C Commands directed to the unit or by a device's front panel. AV/C Digital Interface Command Set General 3.0 supplies commands to make unit internal connections. This document introduces a command to determine the source of data regardless of the implementation-dependent connection configurations.

4.2.3 End-to-end connections

An end-to-end connection is a connection between the originating source and the ultimate destination. And end-to-end connection may include internal connections and peripheral connections. The commands in this specification are designed to create end-to-end connections within and between devices.

An end-to-end connection may include one or more internal connections in a device, i.e. a combination of various internal connections may exist between subunits in the device and its unit plug. The following figure summarizes a typical connection model in case a source and a destination are subunit plugs in different AV/C devices.

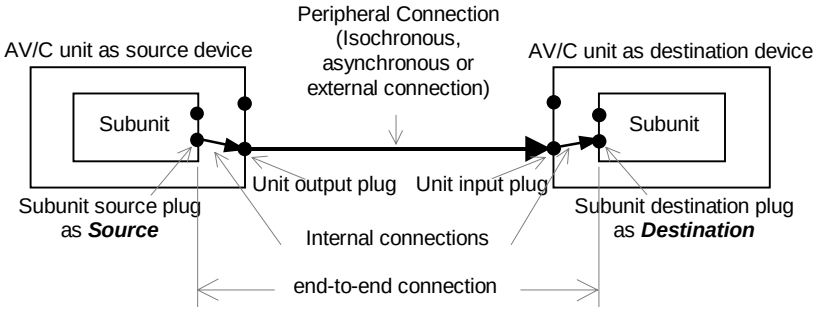


Figure 4.4 – A typical connection model

4.2.4 Sources and destinations

In general, two end points are needed for making one connection and those end points are called the *source* and *destination*. Connections are established between units/subunits using plugs, and plugs are defined by their number and unit or subunit where they are located. Unit plugs serve as a source or destination depending on the type of connection that is being made. For example, in an internal connection, a unit output plug may be a destination, whereas in a peripheral connection, the unit output plug would be a source. The following definitions of source and destination apply to the various connection types.

- **Source**:
  - **Internal connections:** A subunit source plug, or unit input plug (serial bus or external).
  - **Peripheral connections:** A unit output plug (serial bus or external).
  - **End-to-end connection:** A subunit source plug, unit output plug (serial bus or external), or external input plug. In the case of an external input plug, the actual source is the device connected to the external input plug.
- **Destination**:
  - **Internal connections:** A subunit destination plug, or unit output plug (serial bus or external).
  - **Peripheral connections:** A unit input plug (serial bus or external).
  - **End-to-end connection:** A subunit destination plug, unit input plug (serial bus or external), or external output plug. In the case of an external output plug, the actual destination is the device connected to the external output plug.

An AV/C device in which a source or a destination resides is called *source device* or *destination device* respectively. If a source and a destination are reside in the same device, an end-to-end connection between the source and the destination does not include peripheral connections.

4.3 Connection automation

In traditional analog connection models, when a connection is made between devices, each device has to be configured to output and input signals individually. For example, a TV may need to be set to input mode prior to receiving a signal from a camcorder, or a camcorder may need to be set to VTR mode to output signals. Furthermore, there is not an easy way to synch these connections, because a user would have to set each internal device's connections, and then begin the transmission manually.

With 1394 serial bus and AV/C, there is an opportunity to automate the entire connection process and synch the beginning of transmission with one user action. Furthermore, the ability to transition from one

signal format to another in mid-stream becomes possible with commands that support this capability. The commands provided in this specification can be used to realize connection automation such as "one-touch" connections, path change from digital-to-analog and analog-to-digital transmission, and format conversion.

4.4 Operation Model

4.4.1 Connection Initiation

In connection scenarios, user operations are categorized by what a user selects on what device. Basically, it is a controller of a connection that provides some user interface for selecting one or both end points of the connection. The categories are listed as follows.

1) **Source selection:** Select a source on a device that is to be a destination. In this case, the destination acts as a controller to initiate the connection.

2) **Destination selection:** Select a destination on a device that is to be a source. In this case, the source acts as a controller to initiate the connection.

3) **Source and destination selection:** Select both a source and a destination on an external controller. Neither source nor destination initiates the connection.

When the selection is made, the controller performs the connection setup procedure. Required procedure is different by each category and further by each scenario, though connection setup procedure is generally described in chapter 6. When a destination device selects a source, an AV/C command may not be required, since lock transactions are all that are needed to establish the connection.

4.4.2 Conditions under which connections or connection changes occur

There are two conditions under which a connection can occur as follows:

1) A source device may not be transmitting at the time a connection is being made. Under this situation, connection synchronization can be possible. That is, first a connection is made, then a command to begin transmission can be issued to the source device.

2) A source device may already be transmitting at the time a connection is being made. Under this situation, the destination device will receive a signal "in-process".

There are two conditions under which connection changes occur.

1) While connected, a source device may change the format of the data stream. Thus various adjustments in the path and at the destination may have to be made to compensate for the change.

2) While connected, a source device may require more bandwidth for the data stream.

5. Connection Scenarios

5.1 Introduction

At the first stage of 1394 support for consumer digital video devices, the simple operation and procedure for making an isochronous connection between devices was accomplished by setting up a broadcast connection to the default channel. However, as the number of devices on a bus grows, the ability to select a specific device for making a connection with it is necessary. In fact, diversifying capabilities of devices, such as multiple format support, have brought needs for the device selection and effective use of the point-to-point connection into a community of various devices.

In this chapter, connection scenarios are described, i.e. how a user selects devices and how connections are established as the consequence. They are classified according to the analysis in connection initiation described in section 4.4.1. Although most examples introduced in this chapter are for making isochronous connections, the commands in this document also apply to routing to external connections as an alternative. For all connection types, it is beyond the scope of this specification how to discover the capabilities of devices, supported connection types, formats, etc. This will depend on future subunit specifications and other future standards work.

5.2 Connection scenario examples

This section introduces some examples of connection scenarios by each category, which have motivated the design of new commands defined in this specification. At first, examples of basic scenarios according to the three categories are shown in section 5.2.1, 5.2.2 and 5.2.3 respectively. Next, extended scenarios are shown in 5.2.4, and they are special cases of the basic scenarios. All the examples are visualized in Annex B: for the help of understanding.

5.2.1 Source selection

Scenario 1-1: Unit level source selection

A destination device like a TV, a VCR or a printer provides a user interface so that a user can select an AV/C device as a source. When a selection is made, the destination device performs the connection setup procedure as a controller. In the procedure, the controller sets up connection between a unit output plug of the selected source device and a unit input plug of itself, and sets up its necessary internal connections.

It is beyond the scope of this specification how the destination device shows sources to users. For example, some may list up all AV/C devices on the bus while others list up only connectable devices.

Scenario 1-2: Subunit level source selection

A destination device provides user interface so that a user can select a subunit of an AV/C device as a source. When a selection is made, the destination device performs connection setup procedure as a controller. In the procedure, the controller sets up internal connections in the source device, i.e. from a source plug of the selected source subunit to a unit output plug. After that, it performs the same procedure as in the scenario 1-1.

The same scenario can be written in case that an external input plug is selected as a source.

Standardized subunit name is not necessarily shown to users. Any Medium type, icon or some other indirect indication may be used instead.

Scenario 1-3: Content-based selection

This scenario is based on the scenario 1-2 and provides content-based selection using subunit commands. As a consequence, more sophisticated user interface can be provided in a sense. In addition to the procedure described in the scenario 1-2, the destination device, as a controller, may perform configuration of subunit plugs, linking the selected contents to subunit plugs, and pulling a trigger to start transmission.

Scenario 1-4: Using device with limited display capability

This scenario just demonstrates that even a small device can be a controller and perform other scenarios described in this section.

The controller may show candidates one by one in a thin display instead of showing whole list.

5.2.2 Destination selection

Scenario 2: Subunit level destination selection

A source device like an IRD, a VCR or a camera provides a user interface so that a user can select a subunit of an AV/C device as a destination. When a selection is made, the source device performs connection setup procedure as a controller. In the procedure, the controller sets up its necessary internal connections and a connection between a unit output plug of itself and a unit input plug of the selected destination device. Also, the controller sets up internal connections in the destination device, i.e. from the unit input plug to a destination plug of the selected subunit.

The same scenario can be written in case that an external output plug is selected as a destination.

5.2.3 Source and destination selection

Scenario 3: Source and destination simultaneous selection

A device provides a user interface so that a user can select both a source and a destination. When a selection is made, the device performs a connection setup procedure as a controller. In the procedure, the controller sets up internal connections of the selected source device as necessary and a connection between a unit output plug of the selected source device and a unit input plug of the selected destination device. Also, the controller sets up internal connections in the destination device.

5.2.4 Extended scenarios

Scenario 4: Input monitor

This scenario is an extension of scenario 1-1 or 1-2 of the source selection and it demonstrates a kind of familiar operation in analog devices, a so-called input monitor. There is a situation that a selected source device does not output an isochronous stream because it is now receiving an isochronous stream from another device. Nevertheless, a destination device as a controller may still keep the selected source device as a virtual source, and the controller input the isochronous stream that the selected source device is receiving. To realize this operation in an isochronous connection, first the controller must find this situation in the selected source device. This is the concept of "virtual output" (See.4.2.1.1.3). Then, a channel number for the isochronous stream is determined, and then it should establish broadcast-in connection to that channel. The controller should not lock the original connection by overlaying a point-to-point connection to avoid the following situations:

— The overlaid connection prevents other source devices from taking over the channel when the virtual source device is establishing broadcast-in connections from the channel.

— The virtual source device, when changing to another source, must establish a new point-to-point connection that requires bus resources. However, bus resources may be insufficient, particularly for bandwidth, because the overlaid connection is remaining.

Scenario 5: Input monitor for record

The name of this scenario looks like the scenario 4, but the concept and the goal is different. At first, a destination device with recording capability like a VCR or a disc recorder provides user interface so that a user can select a destination that is displaying contents. In most cases, the destination resides in monitoring device like a monitor subunit in a TV or audio subunit in amplifier and does not necessarily have a source plug. When the selection is made, the destination device as a controller first finds a source of the contents, and then it performs a connection setup procedure as it selects the source. If the source resides in another device and it transmits the contents through an isochronous connection, the procedure varies in making isochronous connection between the source device and the controller as follows. A broadcast-in connection will be established by the controller if it is intended to record the contents as is always presented at the monitor device. On the other hand, a point-to-point connection will be overlaid to the connection by the controller if it is intended to record the contents that are displayed at the monitor device at the moment when the user made selection.

Scenario 6: Preset destination selection

Although this scenario is categorized in "destination selection", firstly a destination device makes "source selection". The destination device provides user interface so that a user can select potential source. Namely, the destination device does not perform connection setup procedure but just registers the source. Later, when the user activates the source to output contents, then the source device performs the same connection setup procedure as it selects the destination.

Scenario 7: Change path to destination

This scenario is the only example that a user does not pull the trigger, and it may happen between isochronous and external connections set up by any scenario. When a source device must change connection type for some reason, it performs a connection setup procedure as it selects the destination again. For example, while a VCR without encoder is playing back, it performs this scenario to change path from digital to analog, i.e. isochronous to external, when the tape goes into an analog part from a digital part. Note that the source device and/or the destination device must know that an external connection exists between them to realize this scenario. Then, the destination device is supposed to know which of its external input plugs is connected to the external output plug of the source device.

6. Connection Setup Procedures

This chapter describes connection setup procedures. As mentioned in the previous chapter, the required procedure is different by each category and further by each scenario. However, as a framework, connection setup procedures in several situations are described; such as internal connections, peripheral connections and whole end-to-end connections from a source to a destination.

6.1 Internal connections

A design of internal connections of a device that has multiple plugs and subunits is not standardized. Even if two devices have the same subunits and the same number of plugs, implementations of their internal connections may be different depending on the applications it supports. Therefore, it is not easy to design a controller that controls general internal connections. In order to improve this situation, the following command is designed in addition to the CONNECT AV and CONNECT commands.

— SIGNAL SOURCE command

With this command in control ctype, a controller can request a target to set up internal connections from a source to a destination. In this case, the controller does not care about the exact path; namely the target can arrange the path between the internal source and destination. For example, a controller may send this command to a VCR specifying a source plug of a tuner subunit as a source and a unit output plug as a destination. The purpose of the controller is satisfied as long as the command is accepted and a path is set up either directly or through a Tape Recorder/Player subunit.

On the other hand, if a controller wants to specify a particular path, it can still make connections one by one by sending CONNECT AV or CONNECT commands along the intended path.

6.2 Peripheral connections

As described in section 4.2.1, there are three types of peripheral connections between devices. The procedure to establish each type of connection is different.

6.2.1 Isochronous connections

An isochronous connection is set up between a serial bus output plug of a source device and a serial bus input plug of a destination device. Isochronous connections are either broadcast connections or point-to-point connections. By definition of the reference[R2], any device can establish and manage a point-to-point connection between two devices. However, it is strongly recommended that a destination device establishes and manages a point-to-point connection. The reason is summarized in below.

— Restoration of a point-to-point connection after the bus reset is troublesome if performed by a source or the third device, because it causes a certain amount of asynchronous traffic to find a new phy_ID of the destination device. Contrarily, a destination device can get a new phy_ID of the source from a CIP header of the isochronous packet that the source device keeps transmitting to the same channel during and after the bus reset.

— If a source or a third device establishes a point-to-point connection, a destination device can not break the connection for any reason. Also, another device does not have any chance to replace it with another connection. This is a problem when the device that established the connection does not intend to protect it.

In order to introduce above recommended concept, the following command is designed.

— INPUT SELECT command

With this command in control ctype, a controller can request a target to select a device as a source device and input from it as a destination. In most cases when this command is accepted, the target establishes a point-to-point connection and manages it. Therefore, it can break the point-to-point connection through its front panel or accept another request with this command.

6.2.2 Asynchronous connections

An asynchronous connection is set up between an asynchronous output plug of a producer device and an asynchronous input plug of a consumer device. Any device, as a controller, can establish and manage an asynchronous connection between any two devices. The definition for roles of devices, producer, consumer and controller and necessary procedure is defined in the reference [R8].

NOTE — Since asynchronous connection is beyond the scope of this specification, no additional technical information is described here.

6.2.3 External connection

A user can set up an external connection between an external output plug of a source device and an external input plug of a destination device. In order to realize the scenarios described in section 5.2, though there are some limitations, a device must know that an external connection exists between a source device and a destination device. Then, the INPUT SELECT command can also be used for a setup of an external connection.

If a device requires the use of an external connection, the device may send data out of its plug without knowing whether that plug is connected or not. It is implementation dependent whether a warning to a user is presented as a result of a switch from serial bus plug transmission to external plug transmission.

6.3 End-to-end connection

This section describes the whole procedure for making an end-to-end connection based on scenario 3; source and destination simultaneous selection, because all necessary processes are included as follows.

A. Set up a source device (i.e. have the source device make its internal connections) to be ready for output. This is achieved by the SIGNAL SOURCE command.

B. Set up a connection between unit plugs. This is achieved by the INPUT SELECT command for peripheral connections.

C. Set up a destination device to be ready for input. This is achieved by the INPUT SELECT or by the SIGNAL SOURCE command.

For real-time data transmission through an isochronous or an external connection, it is natural to set up plugs and connections from a source to a destination along the data path in A-B-C order. This is because general AV/C commands that set up plug configurations and connections for real-time data are designed according to criteria that setting configurations in downstream does not affect configurations in upstream while setting or changing configurations in upstream may affect configurations in downstream.

For data transmission through asynchronous connections, both ends should be subunits and it is natural to set up source and destination devices first and then establish a connection between them, i.e. in A-C-B order. This is because data transmission always starts after the end-to-end connection is set up, and plug configurations and the connection is not changed dynamically in an application.

NOTE — Since asynchronous connections are beyond the scope of this specification, the paragraph above is just informative.

In process A - set up a source device to be ready for output, the following setups are done in the order below, as required.

1) Configure a subunit source plug
2) Set up internal connections from the subunit source plug to a unit output plug
3) Configure the unit output plug

Some subunits have a function that configures subunit source plugs; for example, the CONFIGURE command defined in the DISC subunit. In that case, it is essential to configure a subunit source plug, from which an internal connection is set up in the next step. In other subunits, subunit source plugs are configured automatically in accordance with associated contents. Also, there is a case that a unit output plug can be configured by the OUTPUT PLUG SIGNAL FORMAT control command. Although this command configures a unit output plug for a transmission format, it is implied that a target has a function for converting the format of its contents if it accepts this command and changes configuration of the unit output plug.

In process B - set up a connection between unit plugs, only the INPUT SELECT command is issued to make a destination device set up an isochronous point-to-point or broadcast, or external connection. Of course, this command is not required to be sent in other scenarios like source selection because a destination device is the controller.

In process C - set up a destination device to be ready for input, the following setups are done in the order below, as required.

1) Configure a subunit destination plug
2) Set up internal connections from a unit input plug to the subunit destination plug

Some subunits have a function that configures subunit destination plugs; for example again, the CONFIGURE command defined in the DISC subunit. In that case, it is essential to configure a subunit destination plug, to which an internal connection is set up in the next step. In other subunits, subunit destination plugs are supposed to be configured automatically in accordance with receiving contents. In a destination device, a unit input plug is also configured automatically in accordance with receiving contents.

Note that the step 2 of the process C can be done in process B by specifying a destination subunit in the INPUT SELECT command (see 7.2). In this case only, when step 2 in process C is required, it is recommended to accomplish it with process B. This is because combining processes makes the procedure simple and precludes making an unnecessary halfway connection in an error case.

Some examples are visualized in Annex C: to aid in understanding.

7. Unit Commands

This chapter describes unit commands developed for CCM.

The table below summarizes the unit commands for CCM

Table 7.1 – Unit commands

Opcode	Value	Support level (by *ctype*) C	S	N	Comments
SIGNAL SOURCE	$1A_{16}$	M	M	R	Set up a unit internal signal path between plugs regardless of intermediate subunits, and inquire the current status of a unit output plug or a subunit destination plug.
INPUT SELECT	$1B_{16}$	M[1]	M	-	Make a destination device set up a connection between its unit input plug and specified unit output plug of a source device, and inquire the current status of the input.
OUTPUT PRESET	$1C_{16}$	O	O	-	Preset a destination device to a source device, and inquire the current status of a preset.

M = Mandatory, R = Recommended, O = Optional

[1] Detailed support levels are described in section 7.2.1.

7.1 SIGNAL SOURCE command

The SIGNAL SOURCE commands have control, status and notify command type. The SIGNAL SOURCE control command sets up a signal path between plugs in a target regardless of intermediate subunits. The SIGNAL SOURCE status command is used to inquire a signal source and status of a signal flow in a specified destination, which is a unit output plug or a subunit destination. The SIGNAL SOURCE notify command is used to be notified a change of a signal source and/or signal status.

A signal source described here is a plug that is an original source of a signal path in an AV/C unit. Possible signal sources are listed in below.

— A unit input plug (serial bus or external)

— A source plug of a subunit that is generating a signal (Camera subunit, Tuner subunit, etc.)

— A source plug of a subunit that is playing back a signal from a media (Tape Recorder/Player subunit, Disc subunit, etc.)

On the other hands, a subunit source plug is not a signal source when a Tuner subunit outputs a demultiplexed signal from a receiving signal through its demux destination plug, an Audio subunit outputs a processed signal.

7.1.1 SIGNAL SOURCE control command format

The SIGNAL SOURCE control command sets up a unit internal signal path in a target regardless of intermediate subunits (See 6.1) between:

— a subunit source plug and a subunit destination plug

— a subunit source plug and a unit output plug

— a unit input plug and a subunit destination plug

— a unit input plug and a unit output plug

A target may connect these two specified plugs directly, or one or more intermediate subunit(s) may exist between these two plugs.

The format of the SIGNAL SOURCE control command frame is shown in the figure below.

	msb							lsb
opcode	SIGNAL SOURCE ($1A_{16}$)							
operand[0]	reserved				F_{16}			
operand[1]	signal_source							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.1 – SIGNAL SOURCE control command format

The *reserved* field shall be set to 0_{16} and shall not take F_{16} in any future extension unless otherwise the distinction between REJECTED responses for control and status commands is realized by any other way.

signal_source:

The *signal_source* field specifies a source of a signal path in the target.

The encoding of the *signal_source* field is grouped in two as below.

1) The *signal_source* is a unit input plug

If the first byte of the *signal_source* field is FF_{16}, it indicates that the *signal_source* is a unit input plug, and the following byte specifies an input plug number as shown in the figure below.

	msb							lsb
operand[1]	FF_{16}							
operand[2]	source_plug_ID							

Figure 7.2 – signal_source field (signal source is a unit input plug)

The encoding of the *source_plug_ID* field is shown in table below.

Table 7.2 – source plug_ID field

value	source_ plug_ID
00_{16} - $1E_{16}$	Serial bus iPCR[0] - iPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved
$7F_{16}$	Any available Serial bus iPCR
80_{16} - $9E_{16}$	External input plug zero - 30
$9F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available external input plug

2) The *signal_source* is a subunit source plug

If the *signal_source* is a subunit source plug, the first byte of the *signal_source* field specifies a *subunit_type* and a *subunit_ID* of the subunit. The *subunit_type* and *subunit_ID* fields for the source plugs have the same syntax and meaning as an AV/C address (see section 5.3.3 of [R3]) and can support extended *subunit_type* and *subunit_ID*. The last byte of the *signal_source* field specifies a subunit source plug number. The following figure shows the *signal_source* field when there is no *subunit_type* and *subunit_ID* extension.

	msb							lsb
operand[1]	subunit_type					subunit_ID		
operand[2]	source_plug_ID							

Figure 7.3 – signal_source field (signal source is a subunit source plug)

The encoding of the *source_plug_ID* field is shown in table below.

Table 7.3 – source_plug_ID field

value	source_plug_ID
00_{16} - $1E_{16}$	Source plug zero - 30
$1F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available source plug

signal_destination:

The *signal_destination* field specifies a destination of a signal path in a target. The encoding of the *signal_destination* field is the same as the *signal_source* field, but the name and meaning of the fields for plugs are different. The following figures and tables summarize them.

	msb							lsb
operand[1]	FF_{16}							
operand[2]	destination_plug_ID							

Figure 7.4 – signal_destination field (signal destination is a unit output plug)

	msb							lsb
operand[3]	subunit_type					subunit_ID		
operand[4]	destination_plug_ID							

Figure 7.5 – signal_destination field (signal destination is a subunit destination plug)

Table 7.4 – destination_plug_ID field (unit output plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Serial bus oPCR[0] - oPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved
$7F_{16}$	Any available Serial bus oPCR
80_{16} - $9E_{16}$	External output plug zero - 30
$9F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available external output plug

Table 7.5 – destination_plug_ID field (subunit destination plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Destination plug zero - 30
$1F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available destination plug

7.1.2 SIGNAL SOURCE control response format

The format of the SIGNAL SOURCE control response frame is shown in the figure below.

	msb							lsb
opcode	SIGNAL SOURCE ($1A_{16}$)							
operand[0]	reserved					result_status		
operand[1]	signal_source							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.6 – SIGNAL SOURCE control response format

The *reserved* field shall be set to 0_{16}.

result_status:

The *result_status* field provides information about the result after processing the SIGNAL SOURCE control command, as described in the table below.

Table 7.6 – result_status field

value	result_status	response
0_{16}	source	ACCEPTED
1_{16}	not source (through)	ACCEPTED
2_{16} - E_{16}	reserved	-
F_{16}	no information	REJECTED

The “source” means that the target sets up an internal signal path as specified in a SIGNAL SOURCE control command, and the target can make the specified *signal_source* generate a signal and transmit through the path.

The “not source (through)” means that the target sets up an internal signal path as specified in a SIGNAL SOURCE control command, but the target cannot make the specified *signal_source* generate a signal. This occurs only when a subunit source plug is specified as a *signal_source* and the subunit does not generate a signal but just through or process a receiving signal.

signal_source:

The *signal_source* field has the same value as the SIGNAL SOURCE control command except for the case that a controller does not indicate any specific plug and it relies on a target. Then, the *source_plug_ID* field of a SIGNAL SOURCE control command is set to $7F_{16}$ or FF_{16}, and the target shall decide a plug and updates the *source_plug_ID* field in an ACCEPTED response.

signal_destination:

The *signal_ destination* field has the same value as the SIGNAL SOURCE control command except for the case that a controller does not indicate any specific plug and it relies on a target. Then, the *destination_plug_ID* field of a SIGNAL SOURCE control command is set to $7F_{16}$ or FF_{16}, and the target shall decide a plug and updates the *destination_plug_ID* field in an ACCEPTED response.

7.1.3 Handling of SIGNAL SOURCE command

If a target does not have a signal path from a *signal_source* to a *signal_destination* specified in a control command, the target shall return a NOT IMPLEMENTED response.

If a target has a signal path from a *signal_source* to a *signal_destination* specified in a control command but it cannot set up the path at the moment, the target returns a REJECT response to a control command. For example, when a Tape Recorder/Player subunit is playing back and cannot make a signal from a Tuner subunit pass through its source plug nor route through another path, the target rejects SIGNAL SOURCE control command that specifies the Tuner subunit as a source. On the other hand, when a Tuner subunit outputs a demultiplexed signal from a receiving signal through its demux destination plug, the target accepts a SIGNAL SOURCE control command that specifies the Tuner subunit as a source. Then, the target indicates "not source (through)" to the *result_status* (See Table 7.6) in a response. In this case, the original signal is modified through the Tuner subunit, and a controller can check the modification by the *signal_status* field in a SIGNAL SOURCE status command (See Figure 7.10 – signal_status field).

A target shall not return an INTERIM response for a SIGNAL SOURCE control command.

7.1.4 SIGNAL SOURCE status command format

The SIGNAL SOURCE status command is used to inquire a "signal source" and the current status of a specified destination, which is a unit output plug or a subunit destination plug.

The format of the SIGNAL SOURCE status command frame is shown in the figure below.

	msb							lsb
opcode	SIGNAL SOURCE ($1A_{16}$)							
operand[0]	FF_{16}							
operand[1]	FF_{16}							
operand[2]	FE_{16}							
operand[3]	signal_destination							
operand[4]								

Figure 7.7 – SIGNAL SOURCE status command format

signal_destination:

The *signal_destination* field specifies the destination of the signal path in the target. The encoding of the *signal_destination* field of the status command is the same as the SIGNAL SOURCE control command. However, the controller can not specify 0x7F (any available serial bus oPCR) and 0xFF (any available external output plug or any available subunit destination plug) in the *destination_plug_ID* field.

7.1.5 SIGNAL SOURCE status response format

The format of the SIGNAL SOURCE status response frame is shown in the figure below.

	msb							lsb
opcode	SIGNAL SOURCE ($1A_{16}$)							
operand[0]	output_status				conv	signal_status		
operand[1]	signal_source							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.8 – SIGNAL SOURCE status response format

output_status:

The *output_status* field indicates the status of a plug specified in the *signal_destination* field.

Depending on *signal_destination*, the values that are allowed to be set in the *output_status* field are defined below and summarized in Table 7.7, Table 7.8 and Table 7.9.

1) The *signal_destination* is a serial bus oPCR

Table 7.7 – output_status field (signal_destination is a serial bus oPCR)

value	output_status
0_{16}	effective
1_{16}	not effective
2_{16}	insufficient resource
3_{16}	ready
4_{16}	virtual output
5_{16} - 7_{16}	reserved

The "effective" *output_status* means that data packets are flowing through the oPCR.

The "not effective" *output_status* means that signal does not flow into the oPCR because of some reason (such as no internal connection, no data from the *signal_source*).

The "insufficient resource" *output_status* means the oPCR outputs empty packets because of a bus bandwidth shortage.

The "ready" *output_status* means that a signal is flowing to the oPCR, but the oPCR doesn't output packets because no isochronous connection is established.

The "virtual output" *output_status* means that the oPCR virtually outputs the identical signal that is being received by the unit's iPCR. This "virtual output" concept is used to avoid wasted use of the channel and bandwidth. Refer to section 4.2.1.1.3, "Monitor connections and virtual output" for more information about the meaning of virtual output.

When the *output_status* of an oPCR is "virtual output", the oPCR can be either "on-line" or "off-line". If the oPCR is "on-line" and "connected", namely "active", it is recommended to transmit only the empty packets. Another device should not establish a point-to-point connection on an oPCR that is set to virtual output.

The following figure illustrates the decision flow on the target.

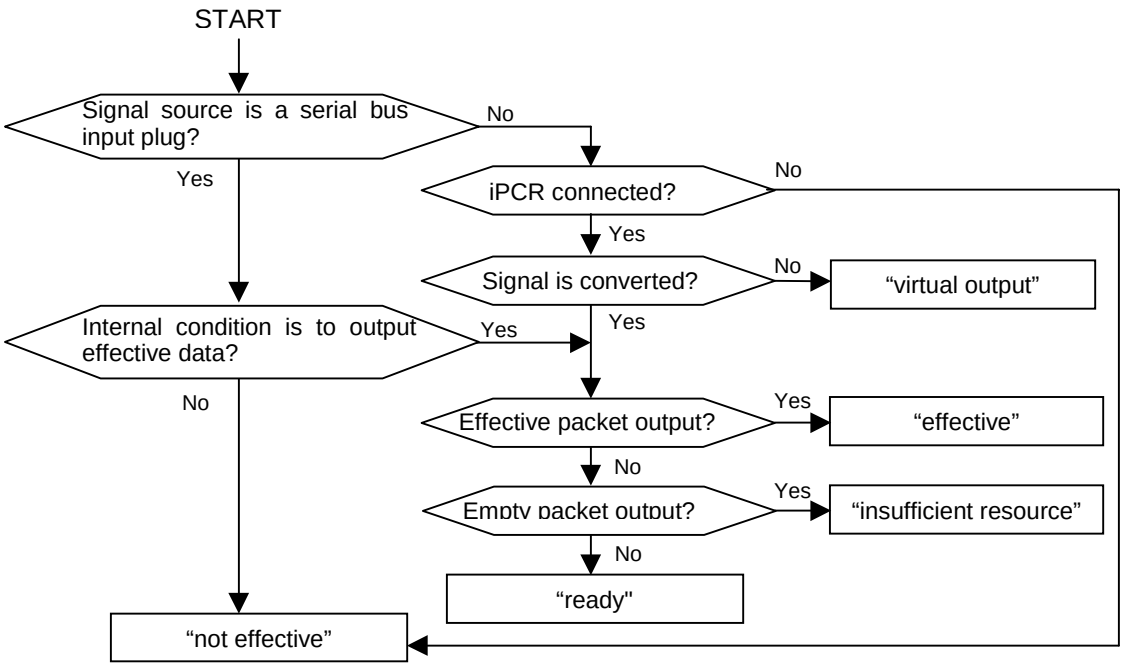


Figure 7.9 – output_status decision on the target

The status that is returned in the *output_status* field of the SIGANAL SOURCE status command denotes the state of a plug in AV/C level. Particularly, the state of a serial bus oPCR is determined by the internal signal flow and isochronous connections that are established on it. To describe this relationship, plug state model is provided in Annex A.2.

2) The *signal_destination* is an external output plug

Table 7.8 – output_status field (signal_destination is an external output plug)

value	output_status
0_{16}	effective
1_{16}	not effective
2_{16} - F_{16}	reserved

The "effective" *output_status* means that the signal flows into the external plug.

The "not effective" *output_status* means that the signal does not flow into the external plug.

3) The *signal_destination* is a subunit destination plug

Table 7.9 – output_status field (signal_destination is a subunit destination plug)

value	output_status
0_{16}	effective
1_{16}	not effective
2_{16} - F_{16}	reserved

The "effective" *output_status* means that the signal flows into the subunit destination plug.

The "not effective" *output_status* means that the signal does not flow into the subunit destination plug.

conv:

The *conv* field indicates whether a format configured to transmit through a *signal_destination* plug can be changed or not. Its encoding is shown in table below.

Table 7.10 – conv field

value	meaning
0	Transfer format can not be changed at the *signal_destination* plug
1	Transfer format can be changed at the *signal_destination* plug

This field is only used when the *signal_destination* indicates a serial bus oPCR, and the format can be changed by the OUTPUT PLUG SIGNAL FORMAT control command (see [R3]). For an external output plug and a subunit destination plug, the *conv* field is set to zero in the response.

For example, if a VCR unit can convert the output signal format on its serial bus output plug from DV to MPEG, or conversely, and if it can be controlled by the OUTPUT PLUG SIGNAL FORMAT control command, the *conv* field is set to one.

signal_status:

The *signal_status* field indicates whether the signal at the *signal_destination* is modified or not from the signal at the *signal_source*, and if so, what kind of modifications are performed. This field consists of four bits, where each bit represents a type of modification in the signal. The bit representations are shown in the figure below.

msb			lsb
processed	filtered	converted	OSD overlaid

Figure 7.10 – signal_status field

An identical *signal_status* is determined when all bits are zero, and indicates that the signal at the *signal_destination* is the same as the signal at the *signal_source*.

When the "processed" bit is one, it indicates that the signal is modified by a means other than filtering, conversion, or via an OSD overlay described below. The manipulation at the Function Block of the audio subunit [R6] would cause this bit to be one.

When the "filtered" bit is one, it indicates that a subset of the original signal has been extracted from the original signal that came from *signal_source*. At the time this specification was developed, the Tuner subunit [R5] filters the signal using its demultiplexing function.

When the "converted" bit is one, it indicates that the signal format at the *signal_destination* has been converted from the signal format at the *signal_source*.

When the "OSD overlaid" bit is one, it indicates that On Screen Display data has been overlaid on the signal.

Any combination of the bits above is possible.

signal_source:

The detail of the *signal_source* field is described below, 1), 2) and 3).

1) The *signal_source* is an isochronous input plug and the *output_status* is "virtual output"

When *signal_source* is a unit input plug and the *output_status* is "virtual output", the *signal_source* field has the structure as shown in the figure below.

	msb							lsb
operand[1]	11_2			isochronous_channel				
operand[2]	source_plug_ID							

Figure 7.11 – signal_source field (*output_status* is "virtual output")

When the *output_status* field is "virtual output", the last six bits of the first byte in the *signal_source* field is the *isochronous_channel* field, and it denotes the isochronous channel, which the target virtually outputs, i.e. the isochronous channel which the target inputs from an iPCR. The following byte is the *source_plug_ID* field and specifies the iPCR. Its encoding is shown in the table below.

Table 7.11 – source_plug_ID field

Value	source_plug_ID
00_{16} - $1E_{16}$	Serial bus iPCR[0] - iPCR[30]
$1F_{16}$ - FF_{16}	Reserved

2) The *signal_source* is an external input plug, or an isochronous input plug and the *output_status* is not "virtual output"

When the *output_status* field is not "virtual output", the first byte of the *signal_source* field indicates that the signal source is a subunit source plug or a unit input plug. If the first byte of the field is FF_{16} , it indicates that the *signal_source* is a unit input plug and the following byte specifies the unit input plug number as shown in the figure and table below.

	msb							lsb
operand[1]	FF_{16}							
operand[2]	source_plug_ID							

Figure 7.12 – signal_source field (signal_source is a unit input plug)

Table 7.12 – source_plug_ID field

Value	source_plug_ID
00_{16} - $1E_{16}$	Serial bus iPCR[0] - iPCR[30]
$1F_{16}$ - $7F_{16}$	Reserved
80_{16} - $9E_{16}$	External input plug zero - 30
$9F_{16}$ - FB_{16}	Reserved
FC_{16} - FE_{16}	Used for other purposes
FF_{16}	Not applicable

3) The *signal_source* is a subunit source plug

If the *signal_source* is a subunit source plug, the first byte specifies its *subunit_type* and *subunit_ID* of the subunit. The *subunit_type* and *subunit_ID* fields for the source plugs have the same syntax and meaning as an AV/C address (see section 5.3.3 of [R3]) and can support extended *subunit_type* and *subunit_ID*. The last byte of the *signal_source* field is the *source_plug_ID* field and specifies a subunit source plug number. The following figure shows the *signal_source* field when there is no *subunit_type* and *subunit_ID* extension.

	msb							lsb
operand[1]	subunit_type					subunit_ID		
operand[2]	source_plug_ID							

Figure 7.13 – signal_source field (signal_source is a subunit source plug)

The encoding of the *source_plug_ID* field is shown in the table below.

Table 7.13 – source_plug_ID field

value	source_plug_ID
00_{16} - $1E_{16}$	Source plug zero - 30
$1F_{16}$ - FF_{16}	Reserved

4) No *signal* source is specified.

There are three cases when no signal source is specified.

If there is no signal path to the *signal_destination*, the *signal_source* field indicates "no signal source" and is encoded as shown in the figure below.

	msb							lsb
operand[1]	FF_{16}							
operand[2]	FE_{16}							

Figure 7.14 – signal_source field (signal_source is "no signal source")

If independent audio and video signal sources are connected with the plug using the CONNECT AV command, the *signal_source* field does not indicate a specific source but indicates "AV convergent" and is encoded as shown in the figure below. In this case, the CONNECT AV status command is used to get details. (See [R3].)

	msb							lsb
operand[1]	FF_{16}							
operand[2]	FC_{16}							

Figure 7.15 – signal_source field (signal_source is "AV convergent")

If the signal at the *signal_destination* is composed of signals from multiple signal sources, the *signal_source* field does not indicate a specific source but indicates "multiple sources" and is encoded as shown in the figure below. In this case, the CONNECT status command is used to get details. (See [R3].)

	msb							lsb
operand[1]	FF_{16}							
operand[2]	FD_{16}							

Figure 7.16 – signal_source field (signal_source is "multiple sources")

signal_destination:

The *signal_destination* field has the same value as the SIGNAL SOURCE status command.

7.1.6 SIGNAL SOURCE notify command format

The SIGNAL SOURCE notify command is used to be notified a change of the *signal_source*, *output_status*, *conv* and *signal_status* of the specified destination. The SIGNAL SOURCE notify command has the same syntax as the SIGNAL SOURCE status command. A notification shall be returned when the response changes.

The notify responses (INTERIM and CHANGED) have the same format as the SIGNAL SOURCE status response frame and indicate the current status of the plug for which the notification was requested.

7.2 INPUT SELECT command

The purpose of the INPUT SELECT control command is to provide the function like an input select button on the front panel of a destination device through AV/C command. It is similar to the input select operations provide by existing analog devices (TV, VCR etc.), but it is extended to the selection of serial bus source devices.

The INPUT SELECT control command is used to make the destination device select a specified device as a source, or connect or disconnect a connection with a specified device.

The INPUT SELECT command also has status command type. The INPUT SELECT status command is used to inquire a status of an input plug of a destination device, i.e. what is currently selected or not.

7.2.1 INPUT SELECT control command format

The format of the INPUT SELECT control command frame is shown in the figure below.

	msb							lsb
opcode	INPUT SELECT ($1B_{16}$)							
operand[0]	subfunction							
operand[1]	reserved							F_{16}
operand[2]	node_ID							
operand[3]								
operand[4]	output_plug							
operand[5]	input_plug							
operand[6]	signal_destination							
operand[7]								
operand[8]	reserved							

Figure 7.17 – INPUT SELECT control command format

The reserved field shall be set to 0_{16} and shall not take F_{16} in any future extension unless otherwise the distinction between REJECTED responses for control and status commands is realized by any other way.

subfunction:

The *subfunction* field determines the operation performed by the target, as defined by the table below.

Table 7.14 – subfunction field

subfunction	value	Support level (ctype = control only)	Action
CONNECT	00_{16}	M[1]	Make the destination device connect to specified output plug of the specified unit.
PATH CHANGE	01_{16}	O	Same as the CONNECT subfunction, but a target is allowed to reject if the target does not select the specified source.
SELECT	02_{16}	R	Request a destination device to select a unit output plug of a specified node as a source. Establishing a connection with the plug is not required upon acceptance of this subfunction.
DISCONNECT	03_{16}	O[2]	Make the destination device disconnect the specified peripheral connection.

[1] It is not mandatory for a special case described in (1) below.
[2] It is mandatory for a special case described in (4) below.

1) CONNECT subfunction

The CONNECT subfunction makes the destination device connect to a specified output plug of a specified unit. The CONNECT subfunction requests the target, as a destination device, to make a peripheral connection from a unit output plug of a source node specified in the command and a unit input plug of itself. When a controller requests a target to establish an isochronous connection immediately, it uses this subfunction.

For example, suppose that an oPCR of a D-VHS (source) is specified as *output_plug* and a destination plug of a monitor subunit in a TV (destination) is specified as the *signal_destination* in the INPUT SELECT control command. When the TV receives a command with the CONNECT subfunction, it immediately establishes an isochronous connection between the oPCR of the D-VHS and a selected iPCR of it, and it establishes an internal path from the selected iPCR to the *signal_destination*. If the *output_status* of the D-VHS's oPCR shown in the SIGNAL SOURCE status response is NOT "virtual output", the TV establishes a point-to-point connection. Otherwise, the TV should establish a broadcast-in (See the description about the *output_status* field in 7.1.5 and 7.2.3 3) in detail). After the establishment of the connection, the TV may select another device or internal tuner for the source of the Monitor subunit by its internal operation or receipt of another INPUT SELECT control command, and then it breaks the isochronous connection.

If the target has no ability to establish an isochronous connection, it does not support this command. A digital still camera may be an example of this target type. If the target has no input select operations for an isochronous connection, it may return NOT IMPLEMENTED response.

2) PATH CHANGE subfunction

The PATH CHANGE subfunction requests the same thing as CONNECT subfunction, but allows a target to reject if the target does not select the specified source. When a target rejects this command, it specifies "not selected" in the *result_status* field of the INPUT SELECT control response (Table 7.18).

The PATH CHANGE subfunction indicates that a specified source changes the output plug for the signal that the target receives.

For example, suppose that a TV (target) has selected a VCR as a source device. The VCR has an oPCR and an external output plug. If the VCR changes an output plug from an oPCR to an external output plug according to the playing signal, the VCR may automatically send an INPUT SELECT control command

with PATH CHANGE subfunction to a target, which is like a preset destination device. If the target has not selected the VCR as a source, this subfunction is useful to make the target decide not to change a source. If the target has selected the VCR as a source and knows external connection to the VCR, it may change the input. On the other hand, the VCR will send an INPUT SELECT control command with CONNECT subfunction upon user operation, and the target will change the input even if it has not selected the VCR as a source.

3) SELECT subfunction

The SELECT subfunction requests a target, as a destination device, to select a unit output plug of a node specified in the command as source of a unit input plug of it. Making an isochronous connection is not required upon acceptance of this subfunction, and relies on an input action of the destination device. Namely, the destination device is responsible for establishing and breaking an isochronous connection according to its internal action, and this helps effective use of bus resources. This subfunction is useful for a device of which input select function allows to change a source device even while it does not use the input stream, see "ready" *status* in Table 7.21.

For example, a Mini-DISC recorder in STOP mode may only select a source without establishing a point-to-point connection upon acceptance of this subfunction because it does not use an input stream while it is in STOP mode. When it becomes REC-PAUSE mode, it will establish a point-to-point connection for preparation of recording, and then it may re-establish a point-to-point connection with another node upon acceptance of another INPUT SELECT command even if the subfunction is SELECT.

4) DISCONNECT subfunction

The DISCONNECT subfunction makes the destination device disconnect a specified isochronous connection. Unlike other subfunctions, an input plug shall be specified. This subfunction is introduced so that a controller can remove an isochronous connection. The support of this subfunction prevents a passive destination device from consuming bus resources unnecessarily and further blocking an establishment of another isochronous connection. Therefore, a device shall support this subfunction if it does not support any command that consequently breaks an isochronous connection established by accepting other subfunction of the INPUT SELECT control command.

For example, a format converter box with no front panel may not support any command to break an isochronous connection established by itself. Such a converter box shall support the DISCONNECT subfunction.

On the other hand, devices such as TV, STB and D-VHS may support commands that consequently break an isochronous connection. For example, a TV breaks an isochronous connection upon selecting an internal tuner as a source or selecting the other device as a source by accepting an AV/C command. Also, VCR breaks an isochronous connection upon selecting an internal tuner as a source, selecting the other device as a source, or starting playback by accepting an AV/C command.

node_ID:

The *node_ID* field indicates a serial bus node_ID of a source device that is to be connected to the target, i.e. destination device. The first byte of the *node_ID* field contains the most significant byte of the node_ID.

NOTE — If a target is not "bridge-aware node", it shall not accept an INPUT SELECT control command specifying a remote bus_ID in the *node_ID* field and return a NOT IMPLEMENTED response. The term "bridge-aware node " denotes a future 1394 node that is compliant to the 1394 bridge specification when it is available. Therefore, if a target is not "bridge-aware node", it also returns a NOT IMPLEMENTED response to an INPUT SELECT specific inquiry command specifying a remote bus_ID in the *node_ID* field. Note that a target simply determines a response by a remote or local bus_ID to the specific inquiry command, and does not consider a phy_ID because it is changed dynamically according to the bus configuration.

output_plug:

The *output_plug* field indicates a plug of the source device to be connected to an input plug of the target. A controller shall specify the valid output plug value in this field.

Table 7.15 – output_plug field

value	output_plug
00_{16} - $1E_{16}$	Serial bus oPCR[0] - oPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved
$7F_{16}$	Not applicable
80_{16} - $9E_{16}$	External output plug zero - 30
$9F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available external output plug

For further explanation, refer to the description about *input_ plug* field below.

input_plug:

The *input_plug* field specifies an input plug to be connected on a target. Except for the DISCONNECT subfunction, the value FF_{16} is always used in the control command, and the target selects the input plug. If the target accepts the control command, it updates this field with the selected plug number in an ACCEPTED response.

In the case that a serial bus oPCR of a source device is specified in the *output_plug* field, the INPUT SELECT control command is used to make the destination device establish an isochronous connection between the oPCR and a serial bus iPCR of the target. If the target supports multiple iPCRs, the target selects which iPCR is to be connected.

In the case that an external output plug of a source device is specified in the *output_plug* field, the INPUT SELECT control command is used to make the destination device specify an external input plug that is connected to the external output plug. Therefore, the target is required to know the external connection with the source device and to register the source device. This may be realized by some initial user setting provided by device dependent function.

There is various status of an input plug upon acceptance of an INPUT SELECT control command. The behaviors of a target are described in section 7.2.3.

signal_destination:

The *signal_destination* field specifies the destination of the signal in a destination device. This may be a subunit destination plug or a unit output plug. The signal destination may or may not be specified.

1) No signal destination is specified

If no signal destination is specified in an INPUT SELECT control command, the *signal_destination* field is described in the figure below.

	msb							lsb
operand[6]	FF_{16}							
operand[7]	FE_{16}							

Figure 7.18 – signal_destination field (no signal destination is specified)

If the target accepts the control command, it selects a default signal destination and updates the *signal_destination* field in the ACCEPTED response frame with corresponding codes, the same as the specified case below 2). This specification does not define how to select the default signal destination, i.e. one of plugs that a signal from the *input_plug* is carried, and it depends on implementation of a target. The same implementation may be applied to a STABLE response for the INPUT SELECT status command described in section 7.2.5.

2) Signal destination is specified

The detail of the *signal_destination* field is described in the figure below.

	msb							lsb
operand[6]	subunit_type					subunit_ID		
operand[7]	destination_plug_ID							

Figure 7.19 – signal_destination field

subunit_type and **subunit_ID:**

The *subunit_type* and *subunit_ID* fields have the same syntax and meaning as the *subunit_type* and the *subunit_ID* in an AV/C address (see section 5.3.3 of [R3]).

The *subunit_type* and *subunit_ID* fields specify a subunit to be a destination in the target. The target may connect the input plug and the destination plug of the destination subunit directly, or one or more intermediate subunit(s) may exist between these two plugs.

The *subunit_type* and *subunit_ID* fields can support extended *subunit_type* and *subunit_ID*. The last byte of the *signal_destination* field specifies a subunit destination plug number. When extended *subunit_type* and/or extended *subunit_ID* is applied in the *signal_destination* field, the one byte *reserved* field shall follow this field.

destination_plug_ID:

If the *subunit_type* and *subunit_ID* is FF_{16}, the *destination_plug_ID* field indicates a unit output plug number and is encoded as Table 7.16. A controller may not select any specific plug and rely on the target. Then the controller sets $7F_{16}$ or FF_{16} to the *destination_plug_ID* field of an INPUT SELECT control command. If the target accepts the control command, it selects an external or serial bus output plug and updates this field in the ACCEPTED response. If the *subunit_type* and *subunit_ID* fields are not equal to FF_{16}, the *destination_plug_ID* field indicates a subunit destination plug number and is encoded as Table 7.17. A controller may not select any specific plug and rely on the target. Then the controller sets FF_{16} to the *destination_plug_ID* field of an INPUT SELECT control command. If the target accepts the control command, it selects a destination plug of the subunit and updates the *destination_plug_ID* field in the ACCEPTED response.

Table 7.16 – destination_plug_ID field (unit output plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Serial bus oPCR[0] - oPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved
$7F_{16}$	Any available Serial bus oPCR
80_{16} - $9E_{16}$	External output plug zero - 30
$9F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available external output plug

Table 7.17 – destination_plug_ID field (subunit destination plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Destination plug zero - 30
$1F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available destination plug

7.2.2 INPUT SELECT control response format

The format of the INPUT SELECT control response frame is shown in the figure below.

	msb							lsb
opcode	INPUT SELECT ($1B_{16}$)							
operand[0]	subfunction							
operand[1]	reserved						result_status	
operand[2]	node_ID							
operand[3]								
operand[4]	output_plug							
operand[5]	input_plug							
operand[6]	signal_destination							
operand[7]								
operand[8]	reserved							

Figure 7.20 – INPUT SELECT control response format

subfunction:

The *subfunction* field has the same as the INPUT SELECT control command.

result_status:

The *result_status* field is set to all one by the controller in the control frame. The target updates it in the response frame to indicate the result of the operation. Detail of the *result_status* field is described in the table below. Response code and status is decided by the flow illustrated in Figure 7.21.

Table 7.18 – result_status field

value	result_status	response
0_{16}	no error	ACCEPTED
1_{16}	disabled	REJECTED
2_{16}	locked	REJECTED
3_{16}	p-to-p (not owner)	REJECTED
4_{16}	insufficient resource	REJECTED
5_{16}	source not found	REJECTED
6_{16}	not selected	REJECTED
7_{16}	not registered	REJECTED
8_{16} - D_{16}	reserved	-
E_{16}	any other reason	REJECTED
F_{16}	no information	INTERIM

The "disabled" *result_status* means that the target cannot select or connect the specified source device because of its internal setting that prohibits from accepting the command. For example, user has selected an option "no input select by the external control " from a menu provided by the destination device.

The "locked" *result_status* means that the target cannot select or connect the specified source device because of its internal state that prohibits from changing a source device by the command. For example, a target rejects the command while it is recording.

The "not selected" *result_status* means that the target rejects the INPUT SELECT control command with the PATH CHANGE subfunction because the target does not select the specified source.

The "any other reason" *result_status* indicates the case such that input signal format does not match the capability of the target.

The other values of *result_status* are grouped into two according to the type of plug, i.e. external or serial bus plug.

1) The *input_plug* is external input plug:

The "not registered" *result_status* means that the target has not registered the specified source device. This *result_status* is only used when an external output plug is specified in *output_plug* in a control command.

2) The *input_plug* is serial bus iPCR:

In case that a target receives an INPUT SELECT control command with SELECT subfunction, the target selects a specified node as a source, but it does not established an isochronous connection until it makes use of an input stream. In case that a target receives an INPUT SELECT control command with CONNECT subfunction, the target immediately tries to establish an isochronous connection between specified source device.

Response code and *result_status* is decided by the flow illustrated in the Figure 7.21 – status decision flow on the target, in case that CONNECT or PATH CHANGE subfunction is specified.

The "p-to-p (not owner)" *result_status* means that the target cannot select or connect the specified source device because all iPCRs of the target are occupied by point-to-point connections established by other nodes.

The "source not found" *result_status* means that the target cannot select or connect the specified source device because specified node or plug does not exist.

The "insufficient resource" *result_status* means that the target cannot connect because bandwidth and / or channel can not be allocated.

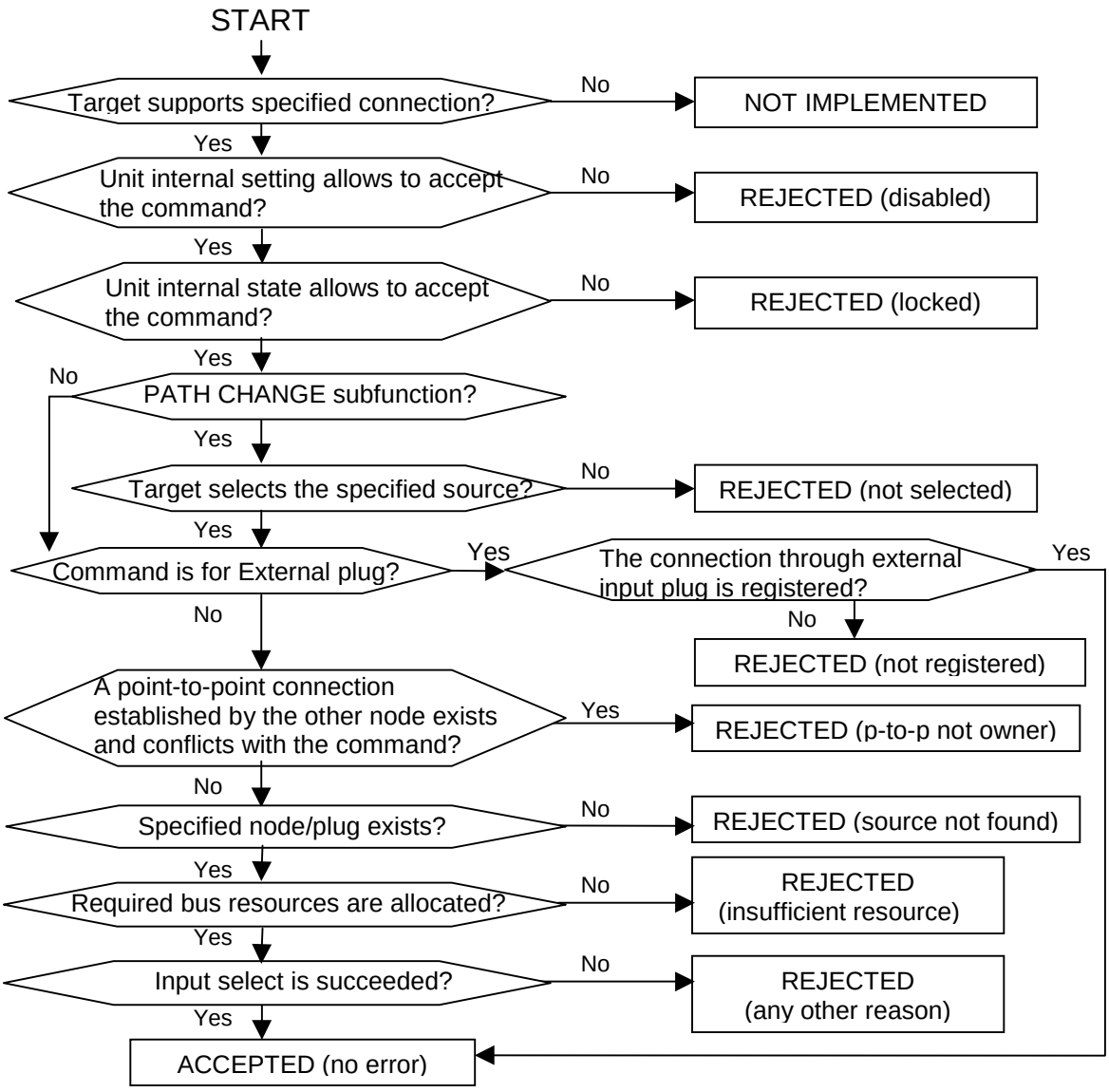


Figure 7.21 – status decision flow on the target

input_plug:

The *input_plug* field specifies the unit input plug to be connected on the target. The selected unit input plug number is updated in a response frame if the control command is ACCEPTED. If the control command is REJECTED or NOT_INPLEMENTED, the same value of the control command returns on the response.

Table 7.19 – input_plug field

value	input_plug
00_{16} - $1E_{16}$	Serial bus plug iPCR[0]-[30]
$1F_{16}$ - $7E_{16}$	Reserved for future specification
$7F_{16}$	Not applicable
80_{16} - $9E_{16}$	External input plug zero - 30
$9F_{16}$ - FD_{16}	Reserved for future specification
FE_{16}	Invalid
FF_{16}	Not applicable for ACCEPTED response

node_ID:

The *node_ID* field has the same value as the INPUT SELECT control command.

output_plug:

The *output_plug* field has the same value as the INPUT SELECT control command.

There is a special case that an oPCR specified in an INPUT SELECT control command is in "virtual output" status. In this case, various behaviors are assumed based on target's implementation, and they are described in section 7.2.3.

signal_destination:

The *signal_destination* field specifies the destination of the signal in a destination device. This may be a subunit destination plug or a unit output plug.

As described in section 7.2.1, a controller may not specify a signal destination or may only specify a subunit or a type of unit output plug in an INPUT SELECT control command. Then, if the target accepts the control command, it selects a signal destination and updates all unspecified parts of this field in the ACCEPTED response. How a signal destination is selected and the encoding is described in section 7.2.1.

There is a case that multiple signal destinations are connected to a unit input plug that a target uses for accepting an INPUT SELECT control command. In such a case, the target may not specify one of those signal destinations when no signal destination is specified in the control command, see Figure 7.18. Then, the target fills the signal_*destination* field in an ACCEPTED response as below.

	msb							lsb
operand[6]	FF_{16}							
operand[7]	FD_{16}							

Figure 7.22 – signal_destination field (signal destination is "multiple")

7.2.3 Handling of INPUT SELECT control command

1) AV/C deferred transaction

INTERIM response may be returned for INPUT SELECT control command because it may take more than 100 milliseconds to process a request. Therefore, it is beneficial to define a reasonable time-out value for

the final response to help a controller providing warning to a user. The following summarizes the time-out definitions for this command.

— A target shall return the final response within 5 seconds from INTERIM response. The earlier response is strongly encouraged for application with quick reaction.

— A controller may give up waiting for the final response and may start some error process after 5 seconds from INTERIM response have elapsed.

When a bus reset occurs, a destination device that has been receiving a stream from a selected source device, either having accepted an INPUT SELECT control command or having selected by itself, tries to restore the prior status. Then, as it takes time for the restoration and causes higher traffic, it is recommended a controller not to send an INPUT SELECT status command until 2 second has elapsed after a bus reset.

When a trigger for sending an INPUT SELECT control command is hit by a user operation on a source device, it is recommended that the device does not re-send the same command automatically after receiving a REJECTED response to the first command. This is to avoid a Ping-Pong situation in a case like a user hits triggers one after another on source devices that are preset to send INPUT SELECT control commands to the same destination device.

2) Exclusive handling with existing isochronous connections

Even if a destination device has a function to select a source device and establish an isochronous connection through a serial bus, it can not remove the point-to-point connection established by other device. For example, if a STB establishes point-to-point connection between a VCR and it and controls the VCR to record the signal from it, the point-to-point connection cannot be broken even if the VCR tries to switch an input signal from the other device. In such a case, the VCR rejects an INPUT SELECT control command with the “p-to-p (not owner)” *result_status.*

On the other hand, a point-to-point connection can be broken by decision of a destination device when it is established by the device itself. For example, a Digital TV establishes a point-to-point connection with a VCR by accepting an INPUT SELECT control command from the VCR. In this case, the Digital TV is allowed to break the connection and re-establish another connection with other device upon a local key operation or accepting an INPUT SELECT control command from another device.

Finally, it is assumed that a destination device breaks a broadcast-in connection by accepting an INPUT SELECT control command from other device.

In any case, it may happen that the currently selected source device is specified in an INPUT SELECT control command. Then, the destination device accepts the command, but takes different actions. If there has been a point-to-point connection on the specified output plug of the source device, which is established by other node, the target overlays a point-to-point connection. If the target has already established a point-to-point connection with the specified output plug of the source device before receiving the command, the target does not overlay another point-to-point connection but returns ACCEPTED response.

3) Capability of monitor connection

In general, an INPUT SELECT control command should be REJECTED with "any other reason" *result_status* if the specified oPCR is off-line state, which is defined in IEC 61883-1. However, there is a special case that the *output_status* of an oPCR is "virtual output", which is obtained by SIGNAL SOURCE status command, and various implementation are assumed as target’s behavior to the INPUT SELECT control command that specifies such an oPCR. This depends whether the target has the capability of the monitor connection or not, as follows.

a) The target does not check the *output_status* of the oPCR.

i) The target establishes a point-to-point connection as long as the oPCR is not off-line state and returns ACCEPTED response. However, the target does not receive any effective packet.

ii) The target finds that the oPCR is off-line state and returns REJECTED response with "any other reason" *result_status*.

b) The target checks the *output_status* of the oPCR, then, it does not need to check IEC 61883-1 plug state.

i) (Recommended implementation) If the target sees "virtual output" *output_status*, it establishes a broadcast-in connection and returns ACCEPTED response. This type of target supports the monitor connection.

ii) If the target sees "virtual output" *output_status*, it returns REJECTED response with "any other reason" *result_status*. This type of target does not support the monitor connection.

It is highly recommended that the controller check the *output_status* (see Table 6.1) of an oPCR when it specifies the oPCR in an INPUT SELECT control command. Particularly, a source device, as a controller, is supposed not to send an INPUT SELECT control command with specifying its oPCR while it is "virtual output".

7.2.4 INPUT SELECT status command format

The INPUT SELECT status command is used to inquire the source node_ID and output plug of the stream entering the specified input plug on the target (destination) . The format of the INPUT SELECT status command frame is shown in the figure below.

	msb							lsb
opcode	INPUT SELECT ($1B_{16}$)							
operand[0]	FF_{16}							
operand[1]	FF_{16}							
operand[2]	FF_{16}							
operand[3]	FF_{16}							
operand[4]	FF_{16}							
operand[5]	input_plug							
operand[6]	FF_{16}							
operand[7]	FE_{16}							
operand[8]	reserved							

Figure 7.23 – INPUT SELECT status command format

input_plug:

The *input_plug* field specifies the input plug on the target. Its encoding is shown in the table below.

Table 7.20 – input_plug field

value	input_plug
00_{16} - $1E_{16}$	Serial bus plug iPCR[0]-[30]
$1F_{16}$ - $7E_{16}$	Reserved for future specification
$7F_{16}$	Not applicable
80_{16} - $9E_{16}$	External input plug zero - 30
$9F_{16}$ - FD_{16}	Reserved for future specification
FE_{16}	Invalid
FF_{16}	Not applicable

7.2.5 INPUT SELECT status response format

The format of the INPUT SELECT status response frame is shown in the figure below.

	msb							lsb
opcode	INPUT SELECT ($1B_{16}$)							
operand[0]	FF_{16}							
operand[1]	status				F_{16}			
operand[2]	node_ID							
operand[3]								
operand[4]	output_plug							
operand[5]	input_plug							
operand[6]	signal_destination							
operand[7]								
operand[8]	reserved							

Figure 7.24 – INPUT SELECT status response format

status:

The *status* field indicates the status of a unit input plug specified in the *input_plug* field. Its encoding is shown in the table below.

Table 7.21 – status field

value	status	description
0_{16}	active	A connection is established with a source specified in the *node_ID* field.
1_{16}	ready	A source specified in the *node_ID* field is selected but a connection is not established.
2_{16}	no selection	Any source is not selected.
3_{16}	cannot input	A source specified in the *node_ID* field is selected but has failed to connect and the target is still trying to establish a connection.
4_{16}	insufficient resource	A source specified in the *node_ID* field is selected but has failed because of bus resource shortage and the target is still trying to establish a connection.
5_{16} - F_{16}	reserved	reserved

The “active” status means that the target (destination) device has selected a source device and established a connection on the input plug.

The “ready” status means that the target (destination) device has selected a source device but no connection is established on the input plug.

The “no selection” status means that the target (destination) device does not select any source device and no connection is established on the input plug.

The “cannot input” status means that the target (destination) device failed to establish a connection from a selected source device and is still trying.

The “insufficient resource” status means that the target (destination) has failed to establish a connection from a selected source device because the target could not get bandwidth and/or channel and is still trying.

NOTE — While a target is trying to establish a connection and has not detected any error, the target may return an IN TRANSITION response with “active” status.

The status that is returned in the *status* field of the INPUT SELECT status command denotes the state of an input plug in AV/C level. Particularly, the state of a serial bus iPCR is determined by the internal setting of input selection and signal flow, and isochronous connections that are established on it. To describe this relationship, plug state model is provided in Annex A.1.

node_ID:

The *node_ID* field indicates a serial bus node_ID of a source device that is connected to the target, i.e. destination device. The first byte of the *node_ID* field contains the most significant byte of the node_ID.

If only a broadcast-in connection is established on the *input_plug* of which status is inquired, the *node_ID* is not specified and is set to $FFFF_{16}$ in the response. Then, *output_plug* is also not specified and set to FE_{16}. However, both fields are set to valid value when the target device is intentionally performing a monitor connection from a source device which output plug is on "virtual output" status. On the other hands, those fields are not specified when the target is on "no selection" status.

output_plug:

The *output_plug* field indicates the unit output plug that is connected to the input plug of the target specified by the *input_plug* field. When the unit output plug is not specified, the *output_plug* field is set to FE_{16}.

input_plug:

The *input_plug* field has the same value as the INPUT SELECT status command.

signal_destination:

The *signal_destination* field specifies the destination of the signal in a destination device. This may be a subunit destination plug or a unit output plug.

The detail of the *signal_destination* field is shown in the figure below.

	msb							lsb
operand[6]	subunit_type					subunit_ID		
operand[7]	destination_plug_ID							

Figure 7.25 – signal_destination field

subunit_type and **subunit_ID:**

The *subunit_type* and *subunit_ID* fields have the same syntax and meaning as an AV/C address (see section 5.3.3 of [R3]) and can support extended *subunit_type* and *subunit_ID*. The last byte of the *signal_destination* field is the *destination_plug_ID* field.

When extended *subunit_type* and/or extended *subunit_ID* is applied in the *signal_destination* field, the one byte *reserved* field shall follow this field.

destination_plug_ID:

If the first byte of the *signal_destination* field is FF_{16}, the *destination_plug_ID* field indicates a unit output plug number. If the first byte of the *signal_destination* field is not equal to FF_{16}, the *destination_plug_ID* field indicates a subunit destination plug number. The following tables show encoding of the *destination_plug_ID* field for these two cases.

Table 7.22 – destination_plug_ID field (unit output plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Serial bus oPCR[0] - oPCR[30]
$1F_{16}$ – $7F_{16}$	Reserved
80_{16} - $9E_{16}$	External output plug zero - 30
$9F_{16}$ - FC_{16}	Reserved
FD_{16} - FE_{16}	Used for other purposes
FF_{16}	Not applicable

Table 7.23 – destination_plug_ID field (subunit destination plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Destination plug zero - 30
$1F_{16}$ – FF_{16}	Reserved
FF_{16}	Not applicable

When a target returns a STABLE response to an INPUT SELECT status command, the *subunit_type*, the *subunit_ID*, and the *destination_plug_ID* of a destination that receives a signal from the *input_plug* are updated

There are four cases in response.

1) Single destination

If there is only one destination, the corresponding codes are updated in the *signal_destination* field.

2) Return a default destination from multiple destinations

A target may return a default signal destination, i.e. one of plugs to which a signal from the *input_plug* is carried. This specification does not define how to select the default signal destination, and it depends on implementation of a target.

3) Return as multiple

A target may not return any specific signal destination, but indicates “multiple” by encoding as shown in the figure below.

	msb							lsb
operand[6]	FF_{16}							
operand[7]	FD_{16}							

Figure 7.26 – signal_destination field (signal destination is "multiple")

4) No destination

If there is no destination, i.e. “no selection” status, the target returns the codes as shown in the figure below.

	msb							lsb
operand[6]	FF_{16}							
operand[7]	FE_{16}							

Figure 7.27 – signal_destination field (no signal destination specified)

7.3 OUTPUT PRESET command

The OUTPUT PRESET commands have control and status command types.

7.3.1 OUTPUT PRESET control command format

The OUTPUT PRESET control command is used to allow a target (source) device to preset a destination device that is to receive the INPUT SELECT control command.

Source devices that support this command contain one or more preset entries that store destination node_IDs and their plug numbers for future connection using the INPUT SELECT command.

For example, if the target VCR is not capable of selecting the destination device, this command is useful to preset the destination device. A controller may send this command so that the VCR memorizes a destination device to which it sends the INPUT SELECT control command when it requests to input or notifies path change.

The format of the OUTPUT PRESET control command frame is shown in the figure below.

	msb							lsb
opcode	OUTPUT PRESET ($1C_{16}$)							
operand[0]	0	preset_entry_number						
operand[1]	destination_node_ID							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.28 – OUTPUT PRESET control command format

preset_entry_number:

The *preset_entry_number* field identifies a preset entry number. A target manages the numbering of the *preset_entry_number* field, and upon accepting a request for a preset, it assigns one of the unused entries. When a controller tries to make a new preset entry, it specifies the invalid value $7F_{16}$ in the *preset_entry_number* field of the command frame. If there is a free entry, the target returns that entry's number in the response frame, which is between zero and the maximum number that the target supports. For more information, see 7.3.3.

When the controller cancels the preset entry, its number is specified in the *preset_entry_number* field and the *destination_node_ID* and *signal_destination* fields are set to $FFFF_{16}$ in the OUTPUT PRESET control command. The controller that has made a preset entry can only cancel the preset entry.

destination_node_ID:

The *destination_node_ID* field specifies the node ID of a destination device to which a target sends an INPUT SELECT control command when it requests input.

NOTE — If a target is not "bridge-aware node", it shall not accept an OUTPUT PRESET control command specifying a remote bus_ID in the *destination_node_ID* field and return a NOT IMPLEMENTED response. The term "bridge-aware node " denotes a future 1394 node that is compliant to the 1394 bridge specification when it is available. Therefore, if a target is not "bridge-aware node", it also returns a NOT IMPLEMENTED response to an OUTPUT PRESET specific inquiry command specifying a remote bus_ID in the *destination_node_ID* field. Note that a target simply determines a response by a remote or local bus_ID to the specific inquiry command, and does not consider a phy_ID because it is changed dynamically according to the bus configuration.

signal_destination:

The *signal_destination* field specifies the destination of the signal in a destination device specified in the *destination_node_ID* field.

The *signal_destination* field is optional, but used for the *signal_destination* field of the INPUT SELECT control command when it is sent, even if the *signal_destination* field is not specified in the OUTPUT PRESET control command.

1) No signal destination is specified

If the signal destination is not specified, the *signal_destination* field is set as in the figure below.

	msb							lsb
operand[3]	FF_{16}							
operand[4]	FE_{16}							

Figure 7.29 – signal_destination field (no signal destination is specified)

2) Signal destination is specified

The detail of the *signal_destination* field is shown in the figure below.

	msb							lsb
operand[3]	subunit_type					subunit_ID		
operand[4]	destination_plug_ID							

Figure 7.30 – signal_destination field

subunit_type and **subunit_ID:**

The *subunit_type* and *subunit_ID* fields have the same syntax and meaning as an AV/C address (see section 5.3.3 of [R3]) and can support extended *subunit_type* and *subunit_ID*. The last byte of the *signal_destination* field is the *destination_plug_ID* field.

destination_plug_ID:

If the first byte of the *signal_destination* field is FF_{16}, the *destination_plug_ID* field indicates a unit output plug number. If the first byte of the field is not equal to FF_{16}, the *destination_plug_ID* field indicates a subunit destination plug number. The following tables show encoding of the *destination_plug_ID* field for these two cases.

Table 7.24 – destination_plug_ID field (unit output plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Serial bus oPCR[0] - oPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved
$7F_{16}$	Any available Serial bus plug oPCR
80_{16} - $9E_{16}$	External output plug zero - 30
$9F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available external output plug

Table 7.25 – destination_plug_ID field (subunit destination plug)

value	destination_plug_ID
00_{16} - $1E_{16}$	Destination plug zero - 30
$1F_{16}$ - FD_{16}	Reserved
FE_{16}	Invalid
FF_{16}	Any available destination plug

7.3.2 OUTPUT PRESET control response format

The format of the OUTPUT PRESET control response frame is shown in the figure below.

	msb							lsb
opcode	OUTPUT PRESET ($1C_{16}$)							
operand[0]	0	preset_entry_number						
operand[1]	destination_node_ID							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.31 – OUTPUT PRESET control response format

preset_entry_number:

In an ACCEPTED response, the assigned preset entry number is returned in the *preset_entry_number* field.

destination_node_ID:

The *destination_node_ID* field has the same value as the OUTPUT PRESET control command.

signal_destination:

The *signal_destination* field has the same value as the OUTPUT PRESET control command.

7.3.3 Handling of OUTPUT PRESET control command

The resources for memorizing preset entries are device dependent and are limited in number. The target device shall supply preset entry numbers on a first-come-first-serve basis. If all preset entries are filled, then the target shall reject the OUTPUT PRESET control command for assigning new presets.

When an OUTPUT PRESET control command from a controller is rejected, the controller may inquire the destination preset of an entry by using an OUTPUT PRESET status command. The controller then may show a warning dialog like "Presets full. Please delete a preset destination before inserting a new destination". This dialog encourages a user to cancel a preset entry from a device that has made the entry.

Preset entries are based on node_IDs, and restoration is required when a bus reset has occurred. The following summarizes behaviors of devices after a bus reset.

— A target clears all preset entries

— Prior controllers that owned the preset entries re-send the OUTPUT PRESET control command within 5 seconds after the bus reset

— New controller shall not send an OUTPUT PRESET control command before 5 seconds elapse.

A target shall not return an INTERIM response from an OUTPUT PRESET control command.

7.3.4 OUTPUT PRESET status command format

The OUTPUT PRESET status command is used to inquire the status of a preset entry by the *preset_entry_number*.

The format of the OUTPUT PRESET status command frame is shown in the figure below.

	msb							lsb
opcode	OUTPUT PRESET ($1C_{16}$)							
operand[0]	0	preset_entry_number						
operand[1]	FF_{16}							
operand[2]	FF_{16}							
operand[3]	FF_{16}							
operand[4]	FF_{16}							

Figure 7.32 – OUTPUT PRESET status command format

preset_entry_number:

When a controller inquires the status of a preset entry, it specifies the number of the preset entry in the *preset_entry_number* field.

When a controller inquires the number of preset entries that a target supports, it fills in the *preset_entry_number* field with $7F_{16}$. Then, the controller receives the number in the response frame from the target.

7.3.5 OUTPUT PRESET status response format

The format of the OUTPUT PRESET status response frame is shown in the figure below.

	msb							lsb
opcode	OUTPUT PRESET ($1C_{16}$)							
operand[0]	self	preset_entry_number						
operand[1]	destination_node_ID							
operand[2]								
operand[3]	signal_destination							
operand[4]								

Figure 7.33 – OUTPUT PRESET status response format

self:

The *self* field indicates whether the preset entry is set locally or not. If the *self* field equals one, it indicates that the preset entry is set locally in the target. If the *self* field equals to zero, it indicates that the preset entry is set by another controller, the same node as indicated in the *destination_node_ID* field.

preset_entry_number:

If the *preset_entry_number* is set to $7F_{16}$ in the OUTPUT PRESET status command, the number of preset entries that a target supports, i.e. the maximum number of a preset entry plus one, is returned in the *preset_entry_number* field. Then, all other fields are not updated.

If a supported *preset_entry_number* is specified in the OUTPUT PRESET status command, all other fields are updated according to the status of the preset entry. However, when the entry is not occupied, the *destination_node_ID* and *signal_destination* fields shall stay FF FF_{16}.

destination_node_ID:

The *destination_node_ID* field specifies a destination device to which a target sends an INPUT SELECT control command when it requests input.

signal_destination:

The *signal_destination* field specifies a destination in a destination device specified in the *destination_node_ID* field. The syntax is the same as the *signal_destination* field of the OUTPUT PRESET control command.

This page is left intentionally blank

Annexes

Annex A: Plug State Model

This annex describes a model of the AV/C level plug states and their transitions of unit input and output plug according to the status newly defined in this specification. The status of input plug is shown in the *status* field of the INPUT SELECT status response (see Table 7.21), and the status of output plug is shown in the *output_status* field of the SIGNAL SOURCE status response (see Table 7.7). Those two statuses are not symmetric, since they are shown in different commands that share two aspects of end-to-end connection.

As defined in IEC 61883-1, serial bus iPCR and oPCR have four states, active, ready, suspended and idle. Those states are referred in this model to describe the AV/C level plug states properly, and are denoted as like “PCR-active”.

A.1 Input plug states and transitions

The following figure shows a diagram of a unit input plug states and transitions of a destination device. These states denote the status shown in the *status* field of the INPUT SELECT status response.

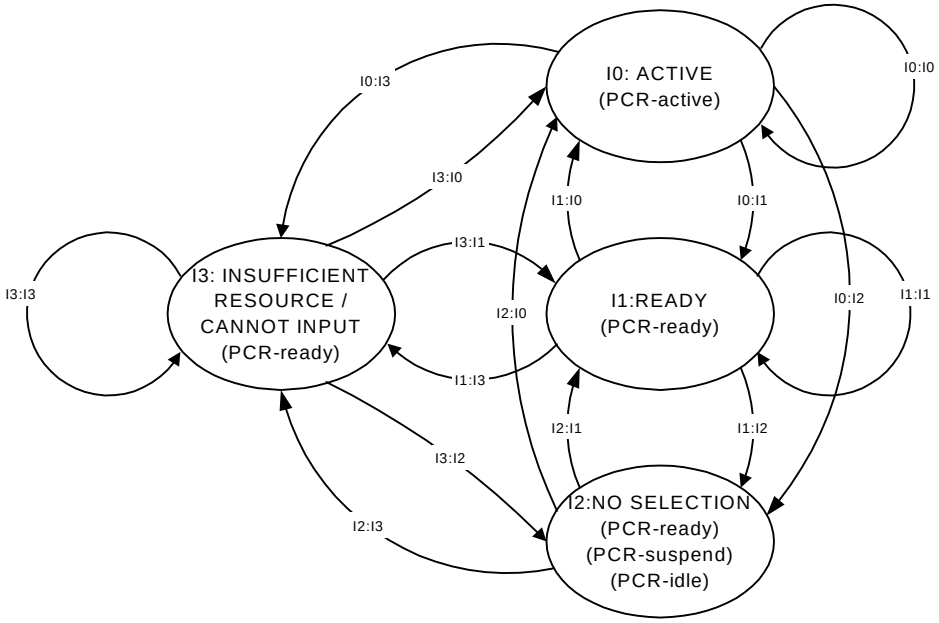


Figure A.1 – State diagram of a unit input plug

Since an external input plug is always in “ACTIVE” state if the destination device selects a source device to input from the plug or in “NO SELECTION” state if it does not select any source device, the following describes with regard to a serial bus input plug.

State I0: ACTIVE. The destination device has selected a source device and established an isochronous connection on the input plug, i.e. PCR-active.

NOTE — The case that the input plug is used for an isochronous connection owned by another device also falls into this status. However, not all information about a source device, i.e. node_ID and plug_ID, may be given in the INPUT SELECT status response because it depends on device implementation.

Transition I0:I0. The destination device changes a source device by the following triggers:

— Local operation of source device selection.

— Acceptance of an INPUT SELECT control command with SELECT or CONNECT subfunction.

Transition I0:I1. The destination device breaks the connection with the selected source device by the following triggers:

— Local operations or acceptance of remote commands that make internal mode deactivate, and break the connection consequently.

— Acceptance of an INPUT SELECT control command with DISCONNECT subfunction.

Transition I0:I2. The destination device breaks the connection and releases the source device by the following triggers:

— Local operations or acceptance of remote commands that invalidate a device selection.

— Acceptance of an INPUT SELECT control command with DISCONNECT subfunction.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the INPUT SELECT status command after this transition.)

Transition I0:I3. The destination device tries to change a source device by the same trigger as transition I0:I0, but fails because of the following reasons:

— Bus resources are insufficient.

— Selected source device does not send any packet from a unit output plug from which the device intended to input because the plug is off-line, i.e. PCR-idle or PCR-suspended.

— Selected source device does not send a stream in a format that the destination device can receive.

— Selected source device has been sending a stream in a speed that the destination device cannot receive.

Also, there is a case that the destination device detects a change in receiving stream and once release the connection because of the following reasons:

— The source device sends only empty packets for a long time because it fails to allocate additional bandwidth or changes to ineffective internal mode.

— The source device does not send any packet because its output plug changes to off-line.

— The source device changes a stream into a format that the destination device cannot receive.

— The source device increases a transmitting speed and the destination device cannot receive. (Broadcast-in only)

State I1: READY. The destination device has selected a source device but no connection is established on the input plug, i.e. PCR-ready.

Transition I1:I0. The destination device establishes an isochronous connection from the selected source device by the following triggers:

— Local operations or acceptance of remote commands that make internal mode activate, and establish an isochronous connection consequently.

— Acceptance of an INPUT SELECT control command with CONNECT subfunction.

Transition I1:I1. The destination device changes a source device by the following triggers:

— Local operation of source device selection.

— Acceptance of an INPUT SELECT control command with SELECT subfunction.

Transition I1:I2. The destination device releases the selected source device by the following triggers:

— Local operations or acceptance of remote commands that invalidate a device selection.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the INPUT SELECT status command after this transition.)

Transition I1:I3. The destination device tries to establish an isochronous connection from selected source device by the same trigger as transition I1:I0, but fails because of the following reasons:

— Bus resources are insufficient.

— Selected source device does not send any packet from a unit output plug from which the device intended to input because the plug is off-line, i.e. PCR-idle or PCR-suspended.

— Selected source device does not send a stream in a format that the destination device can receive.

— Selected source device has been sending a stream in a speed that the destination device cannot receive.

State I2: NO SELECTION. The destination device does not select any source device and no connection is established on the input plug or the input plug is off-line, i.e. PCR-ready, suspend or idle.

NOTE — The case that the off-line input plug is used for an isochronous connection owned by another device also falls into this status. Then, the plug is in PCR-suspend state and no information about a source device is given by the INPUT SELECT status command.

Transition I2:I0. The destination device selects a source device and establishes an isochronous connection from it by the following triggers:

— Local operation of source device selection that forces establishing an isochronous connection.

— Acceptance of an INPUT SELECT control command with SELECT or CONNECT subfunction.

Transition I2:I1. The destination device selects a source device by the following triggers:

— Local operation of source device selection.

— Acceptance of an INPUT SELECT control command with SELECT subfunction.

— Power on. (This may result in selecting a source by default or previous selection.)

Transition I2:I3. The destination device tries to establish an isochronous connection from a selected source device by the same trigger as transition I2:I0, but fails because of the following reasons:

— Bus resources are insufficient.

— Selected source device does not send any packet from a unit output plug from which the device intended to input because the plug is off-line, i.e. PCR-idle or PCR-suspended.

— Selected source device does not send a stream in a format that the destination device can receive.

— Selected source device has been sending a stream in a speed that the destination device cannot receive.

State I3: INSUFFICIENT RESOURCE / CANNOT INPUT. The destination device failed to establish an isochronous connection from a selected source device and is still trying. The input plug is in PCR-ready state.

Transition I3:I0. The destination device succeed to establish an isochronous connection from a source device that it has selected, or it changes a source device and establishes an isochronous connection from it by the following triggers:

— Local operation of source device selection.

— Acceptance of an INPUT SELECT control command with SELECT or CONNECT subfunction.

Transition I3:I1. The destination device changes a source device by the following triggers:

— Local operation of source device selection.

— Acceptance of an INPUT SELECT control command with SELECT subfunction.

Also, there is a case that the destination device gives up trying to establish the connection by the following triggers:

— Local operations or acceptance of remote commands that make internal mode deactivate.

— Acceptance of an INPUT SELECT control command with DISCONNECT subfunction.

Transition I3:I2. The device releases the source device by the following triggers:

— Give up trying to establish an isochronous connection from the source device

— Local operations or acceptance of remote commands that invalidate a device selection.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the INPUT SELECT status command after this transition.)

Also, there is a case that the destination device gives up trying to establish the connection by the following triggers:

— Local operations or acceptance of remote commands that make internal mode deactivate.

— Acceptance of an INPUT SELECT control command with DISCONNECT subfunction.

Transition I3:I3. The destination device changes a source device by the same trigger as transition I3:I0, but fails to establish an isochronous connection from it because of the following reasons:

— Bus resources are insufficient.

— Selected source device does not send any packet from a unit output plug from which the device intended to input because the plug is off-line, i.e. PCR-idle or PCR-suspended.

— Selected source device does not send a stream in a format that the destination device can receive.

— Selected source device has been sending a stream in a speed that the destination device cannot receive.

A.2 Output plug states and transitions

The following figure shows a diagram of a unit output plug states and transitions of a source device. These states denote the status shown in the *output_status* field of the SIGNAL SOURCE status response.

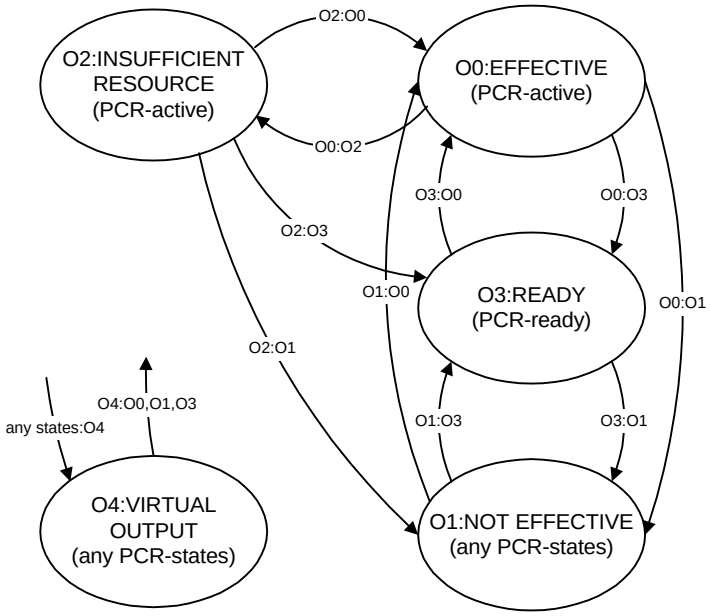


Figure A.2 – State diagram of a unit output plug

Since an external output plug is always in “EFFECTIVE” state if it can output a signal or in “NOT EFFECTIVE” state if it cannot output any signal, the following describes with regard to a serial bus output plug.

State O0: EFFECTIVE. One or more isochronous connections are established on the output plug, i.e. PCR-active state, and the device transmits effective packets from it. The effective packets mean a continuous isochronous stream that carries contents.

Transition O0:O1. The source device becomes a mode unable to output effective packets by the following triggers:

— Internal connection from a subunit source plug or a unit input plug to the output plug that has carried an effective internal signal is removed.

— A subunit source plug or a unit input plug connected to the output plug stops to provide an effective internal signal.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the SIGNAL SOURCE status command after this transition.)

Transition O0:O2. The source device tries to claim additional bus bandwidth from IRM and to update the payload field on the oPCR register upon change of contents to be output, but it fails because of bandwidth shortage.

Transition O0:O3. All isochronous connections that have been established on the output plug are broken.

Transition O0:O4. The source device starts receiving a signal from another source device and does not just repeat it to another channel in order not to waste bus bandwidth.

State O1: NOT EFFECTIVE. The source device does not output any effective packets from the output plug even if an isochronous connection is established. Any PCR states are possible. A device falls into this state when no internal connection is set up to this output plug, or internal connection is set up but no signal flows through it.

Transition O1:O0. The source device becomes a mode to be able to output effective packets, and at the same time, the device establishes an isochronous connection, supposed to be a broadcast-out connection, on the output plug by the following triggers:

— Internal connection is set up from a subunit source plug or a unit input plug that provides an effective internal signal.

— A subunit source plug or a unit input plug connected to the output plug starts to provide an effective internal signal.

— Power on. (This may result in setup for default internal connection and providing an effective signal to the output plug.)

NOTE — It may occur that the state transfers to the state O2:INSUFFICIENT RESOURCE, if the output plug has been PCR-active and sufficient bandwidth has not been allocated. However, it is recommended that any device does not establish isochronous connections and does not make the output plug PCR-active state while it is in the NOT EFFECTIVE state.

Transition O1:O3. The source device becomes a mode to be able to output an effective signal by same triggers as transition O1:O0.

Transition O1:O4. The source device starts receiving a signal from another source device and does not just repeat it to another channel in order not to waste bus bandwidth.

State O2: INSUFFICIENT RESOURCE. The source device is performing retry of claiming additional bus bandwidth from IRM. Then, the output plug is still in PCR-active state and is transmitting only empty packets.

Transition O2:O0. The source device succeeds to claim additional bus bandwidth from IRM or changes contents that does not require additional bus bandwidth.

Transition O2:O1. The source device becomes a mode unable to output an effective signal by the following triggers:

— Internal connection from a subunit source plug or a unit input plug to the output plug that has carried an effective internal signal is removed.

— A subunit source plug or a unit input plug connected to the output plug stops to provide an effective internal signal.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the SIGNAL SOURCE status command after this transition.)

Transition O2:O3. All isochronous connections that have been established on the output plug are broken.

Transition O2:O4. The source device starts receiving a signal from another source device and does not just repeat it to another channel in order not to waste bus bandwidth.

State O3: READY. The source device can output effective packets from the output plug if an isochronous connection is established, i.e. PCR-ready state.

Transition O3:O0. An isochronous connection is established on the output plug. The connection may be a point-to-point connection, or broadcast-out connection on successful retry.

Transition O3:O1. The source device becomes a mode to be unable to output an effective signal by the following triggers:

— Internal connection from a subunit source plug or a unit input plug to the output plug that has carried an effective internal signal is removed.

— A subunit source plug or a unit input plug connected to the output plug stops to provide an effective internal signal.

— Power off. (In this case, the result of this transition may not be observed if the device simply replies a REJECTED response to the SIGNAL SOURCE status command after this transition.)

Transition O3:O4. The source device starts receiving a signal from another source device and does not just repeat it to another channel in order not to waste bus bandwidth.

State O4: VIRTUAL OUTPUT. The source device suggests other devices to input a signal from the original source device from which it is now receiving a signal. The device itself does not output any effective packets from the output plug even if an isochronous connection is established. Any PCR states are possible. However, it is recommended that any device does not establish isochronous connections and does not make the output plug PCR-active state while it is in the VIRTUAL OUTPUT state.

Transition O4: O0. The source device stops receiving a signal from another source device, and at the same time, it starts transmission by establishing an isochronous connection, supposed to be a broadcast-out connection.

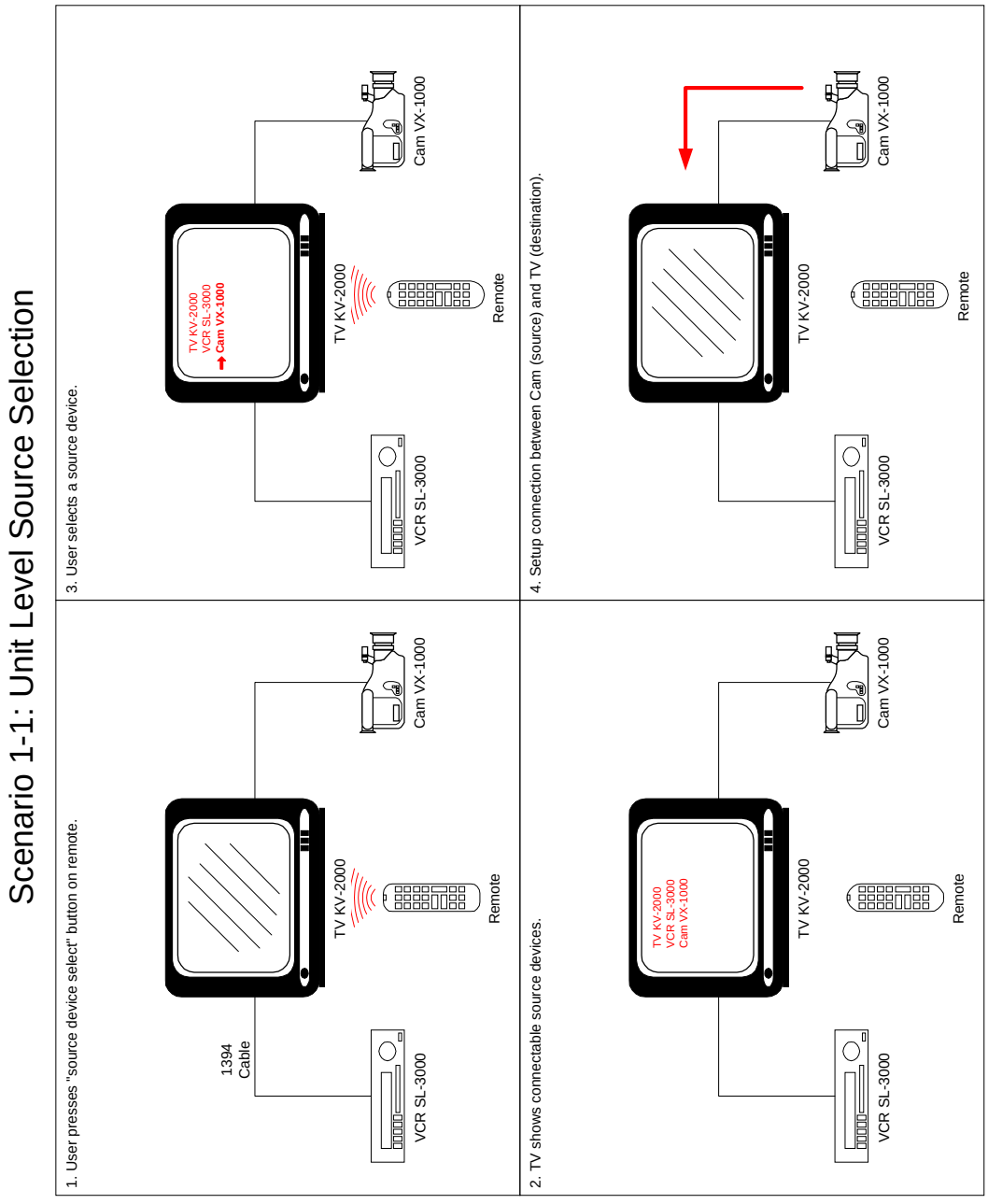
NOTE — It may occur that the state transfers to the state O2:INSUFFICIENT RESOURCE, if the output plug has been PCR-active and sufficient bandwidth has not been allocated. However, it is recommended again that any device does not establish isochronous connections and does not make the output plug PCR-active state while it is in the VIRTUAL OUTPUT state.

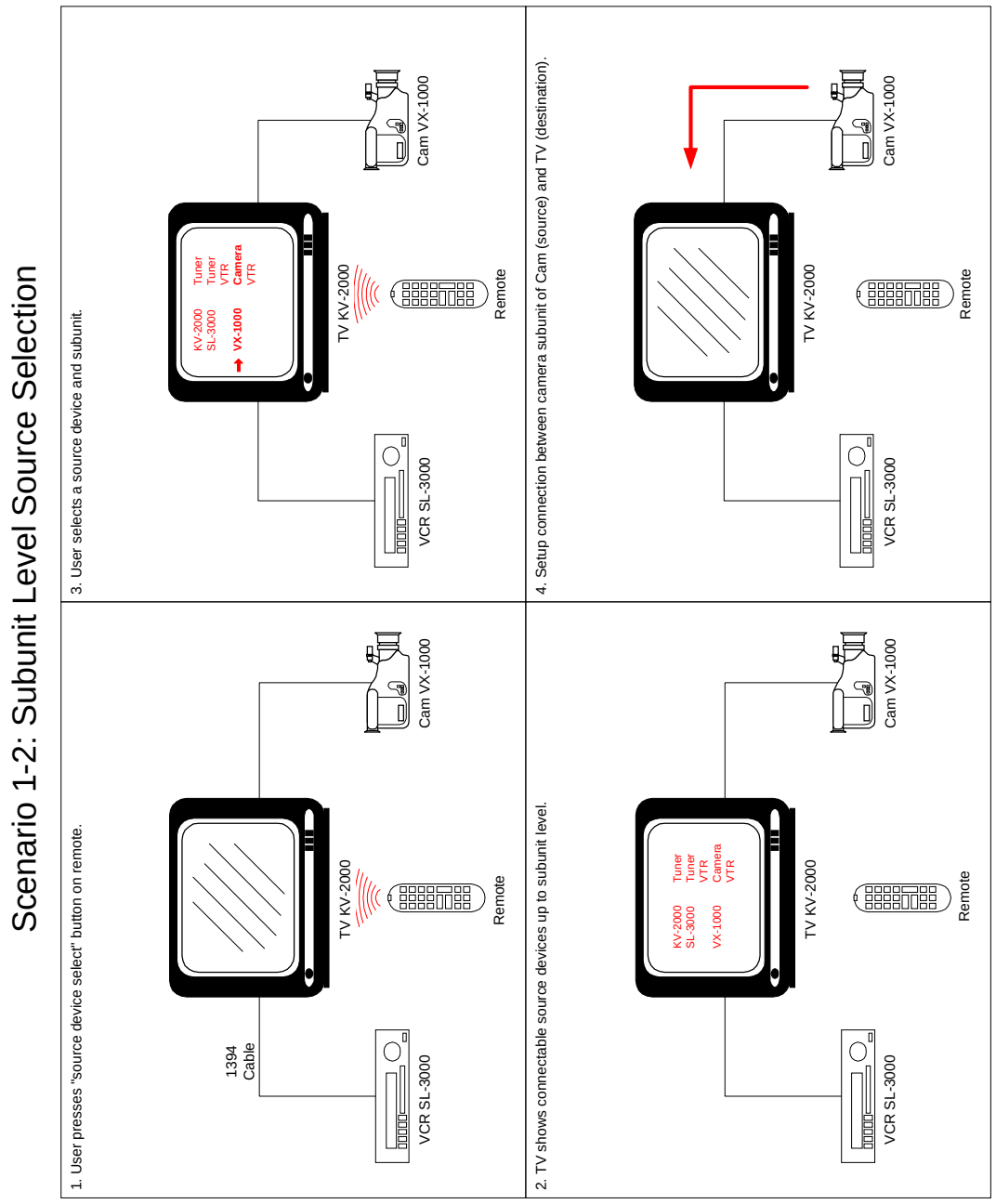
Transition O4: O1. The source device stops receiving a signal from another source device.

Transition O4: O3. The source device stops receiving a signal from another source device, and it becomes ready to output effective packets from the output plug.

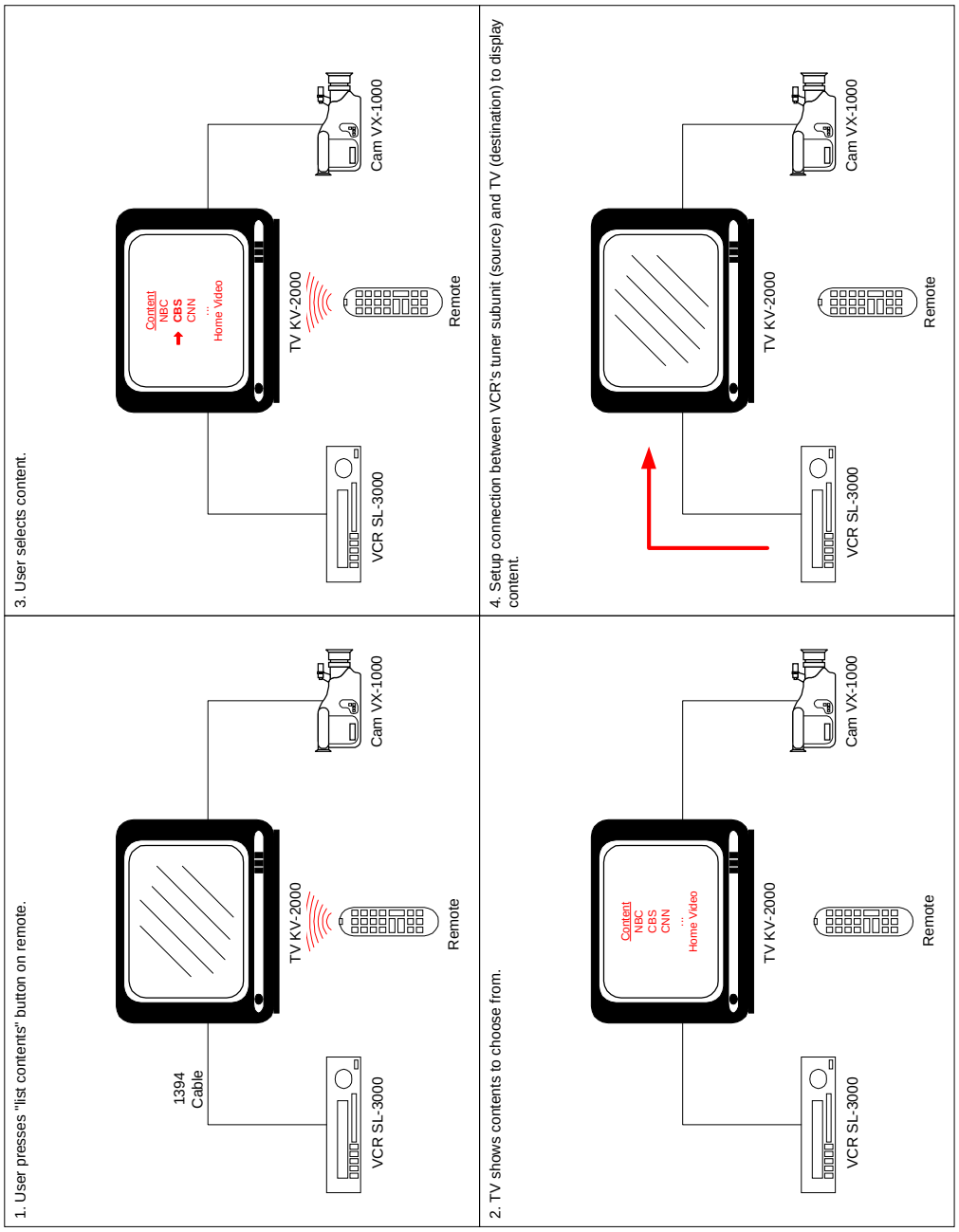
Annex B: Example of Connection Scenarios

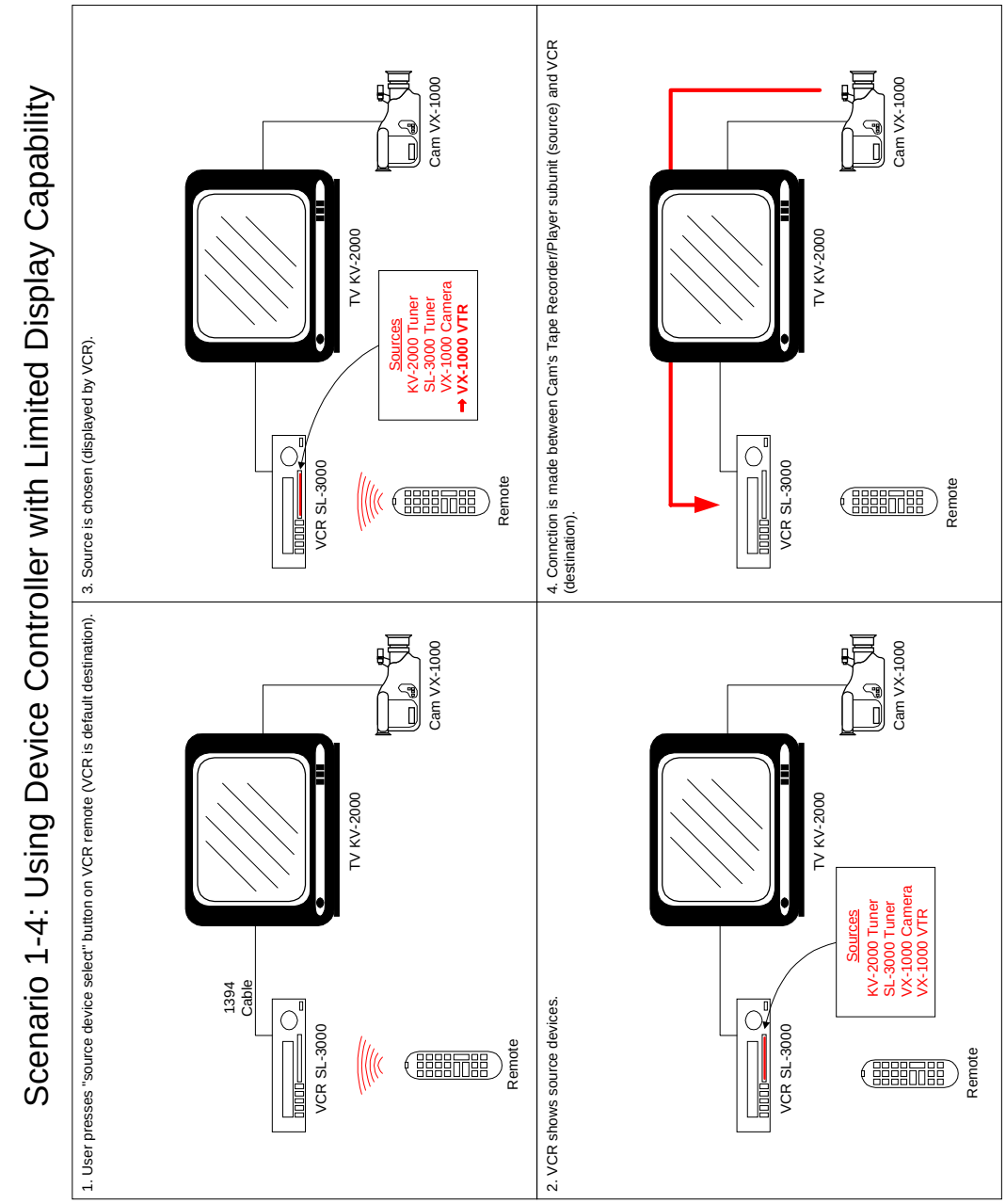
This annex provides examples of connection scenarios. All device model names illustrated in figures are not actual and only examples. Tape Recorder/Player subunit is denoted as “VTR” in display image.

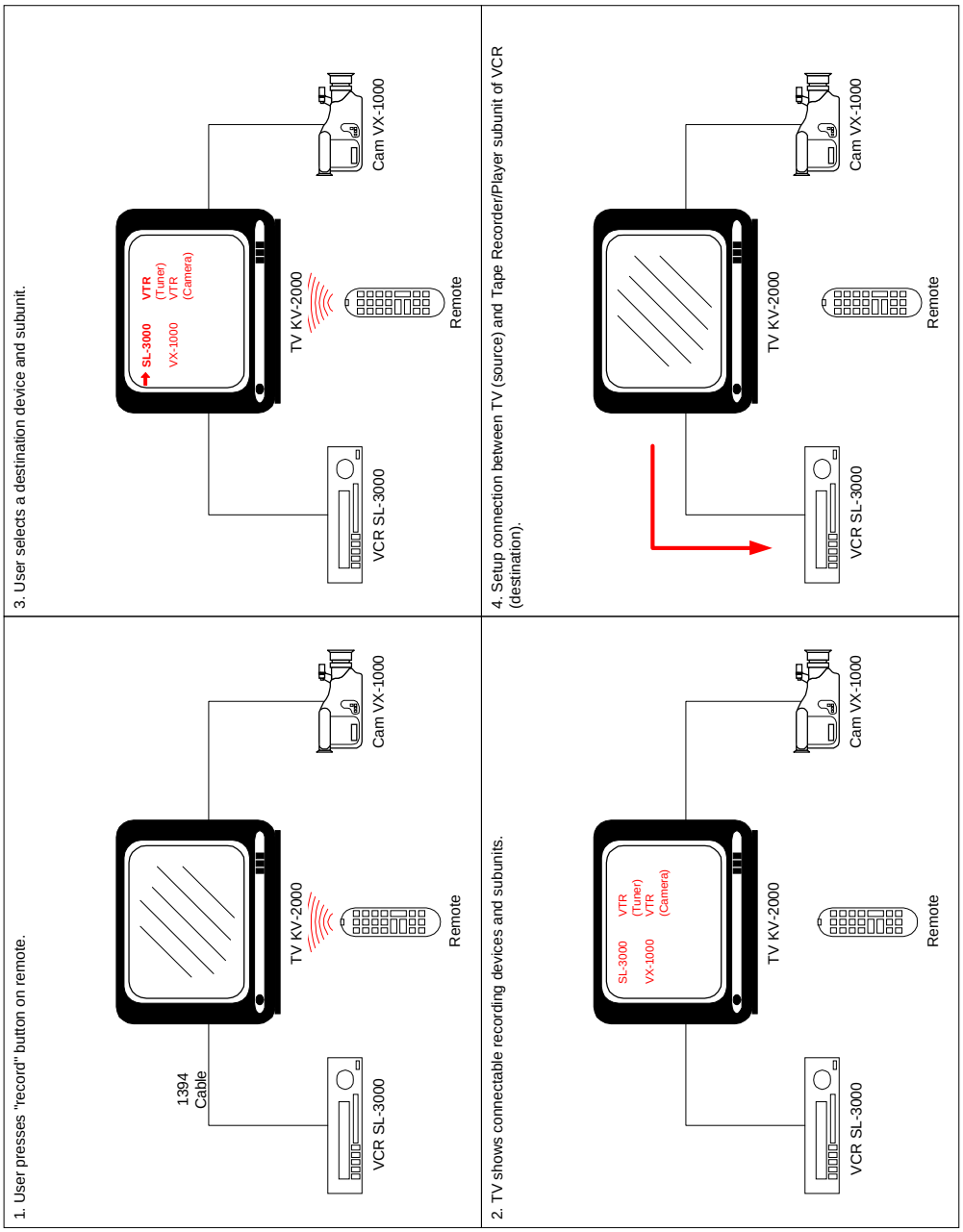




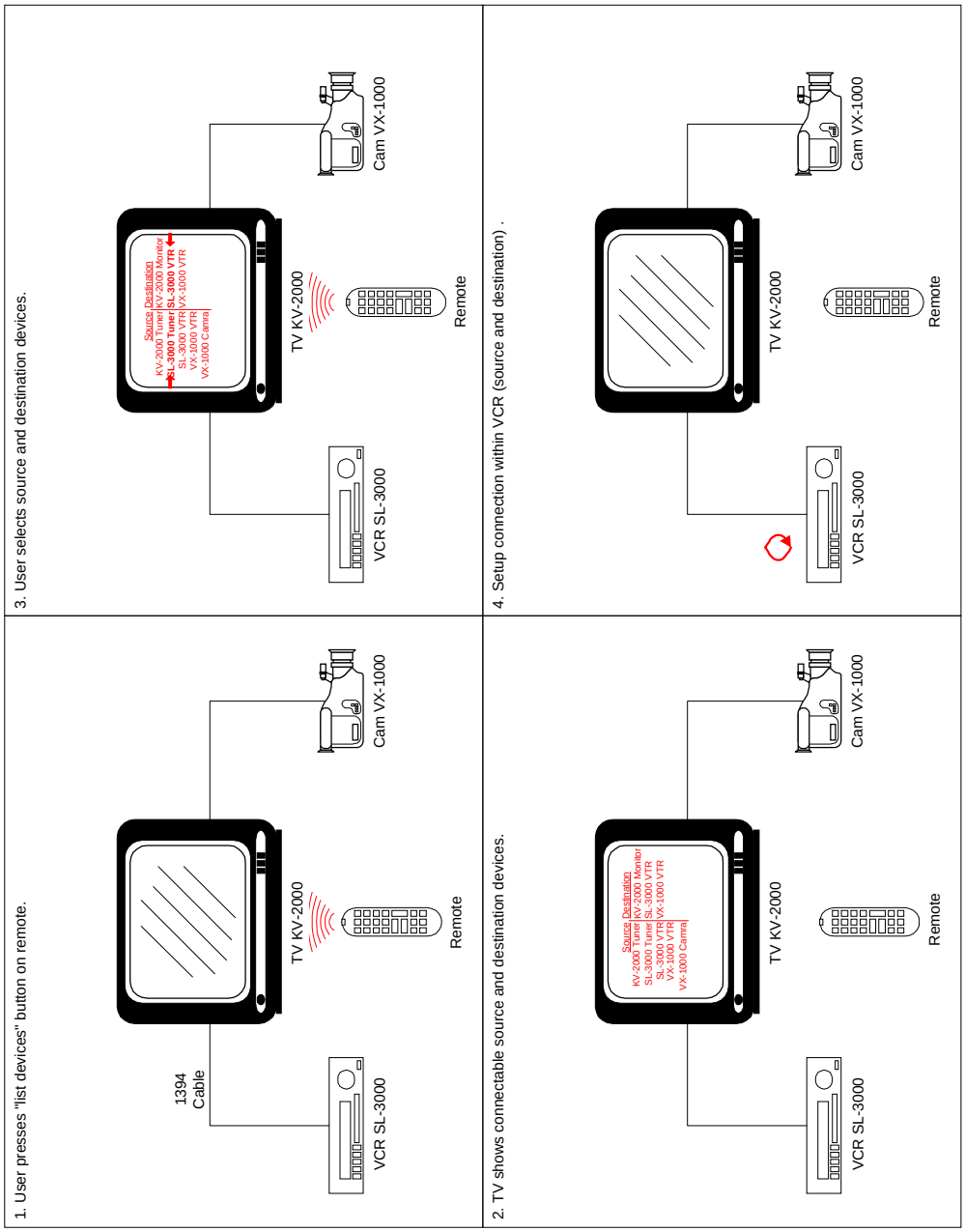
Scenario 1-3: Content-Based Selection

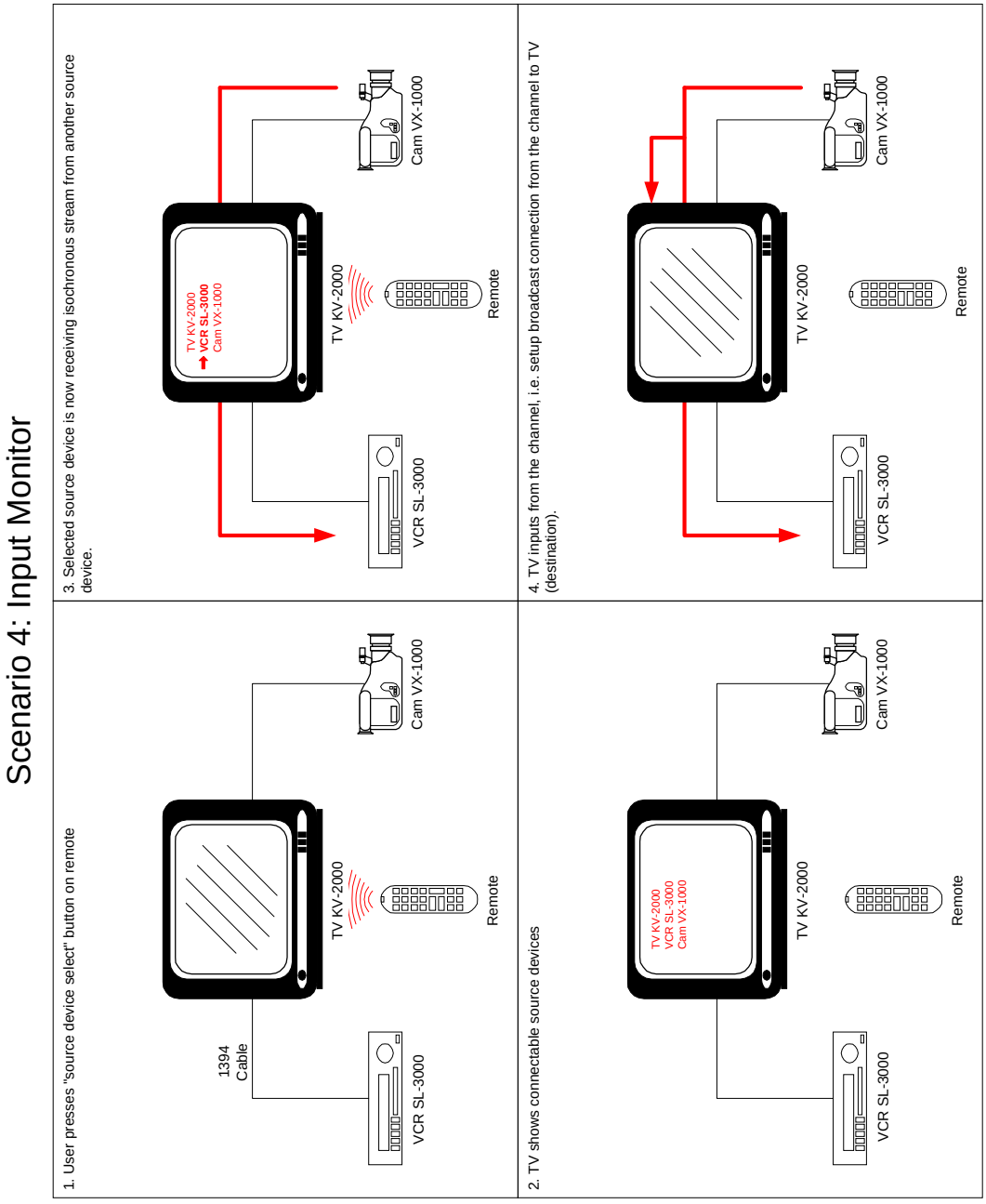




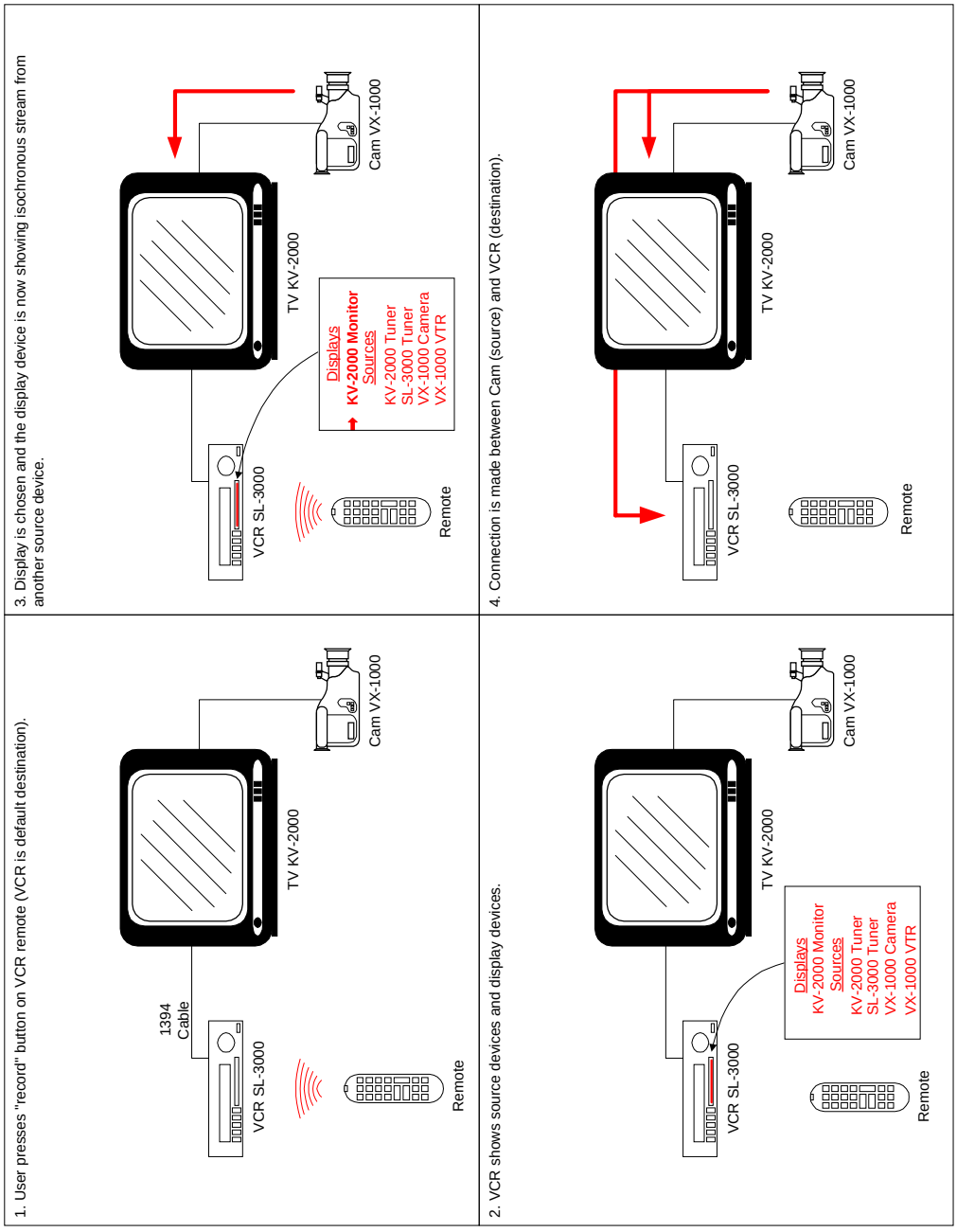


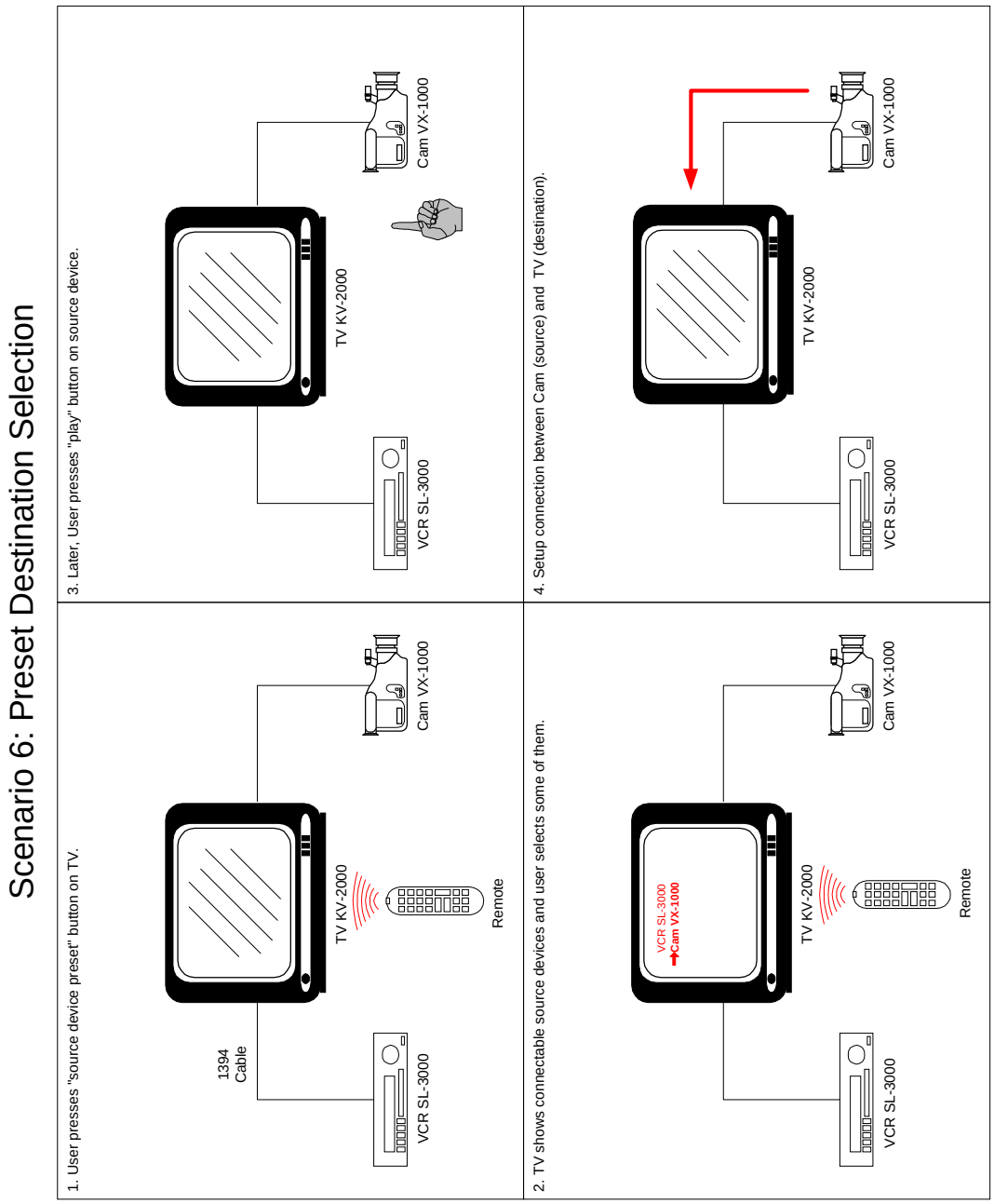
Scenario 3: Source and Destination Simultaneous Selection



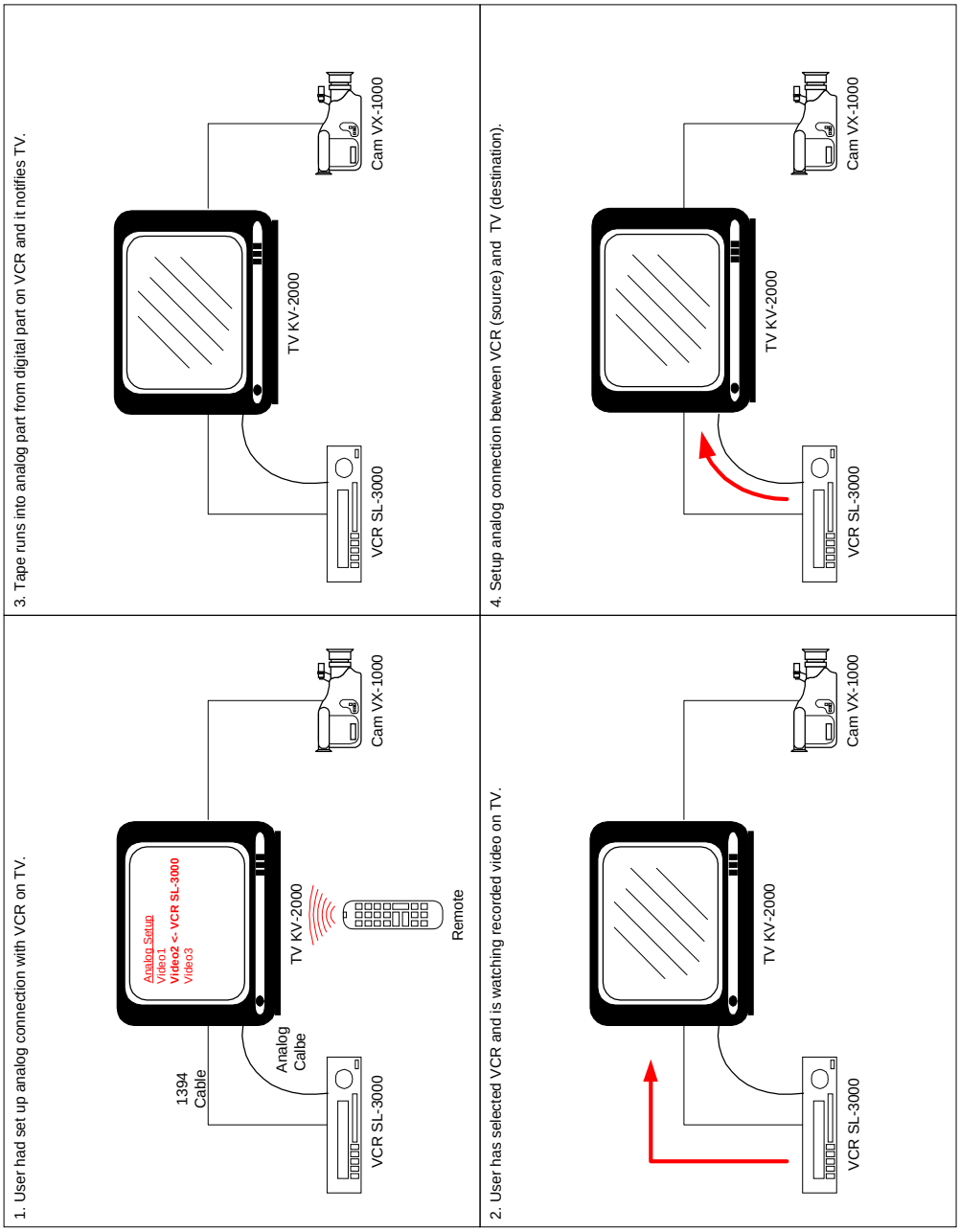


Senario 5: Input Monitor for Record





Scenario 7: Change Path to Destination



Annex C: Example applications

This annex provides examples of practical applications for procedures defined in section 6 ”Connection Setup Procedures” based on scenarios defined in section 5 “Connection Scenarios”..

C.1 Example of a source selection application

Two devices are involved in this application as shown in Figure C.1.

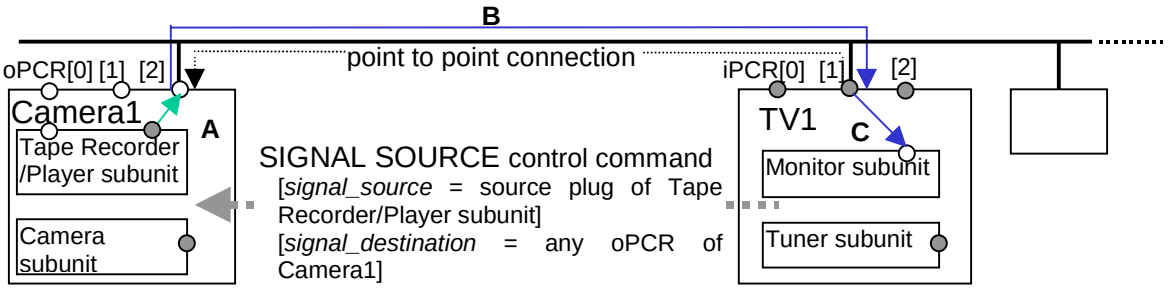


Figure C.1 – Example of a source selection application

TV1 provides a user interface for source device selection and works as a controller. A user selects the Camera1 as a source. Refer to scenario 1-2 also.

The following illustrates a possible procedure performed by the controller, TV1, upon a user operation.

A. Set up a source device to be ready for output.

1 TV1 checks the conditions of the Tape Recorder/Player subunit in Camera1 using a TRANSPORT STATE status command [R9] etc.

2 TV1 checks the internal signal flow and the status of the oPCRs in Camera1 using a SIGNAL SOURCE status commands. An example of command and response frames is shown in the table below.

Table C.1 – Example of SIGNAL SOURCE status command and response frames

field	status command frame	STABLE response frame
output_status	7_{16} (default)	3_{16} (ready)
conv	1_2 (default)	0_2 (can not be changed)
signal_status	F_{16} (default)	0_{16} (identical)
signal_source	$FFFE_{16}$ (default)	38_{16} (Camera subunit, byte 1) 00_{16} (source plug, byte 2)
signal_destination	$FF02_{16}$ (oPCR[2] of Camera1)	<-

3 TV1 sends a SIGNAL SOURCE control command to Camera1.

The *signal_source* and the *signal_destination* fields are specified in this command. If the *destination_plug_ID* is set as "any isochronous plug", Camera1 selects an output plug, e.g. oPCR[2].

1) Camera1 sets up an internal path from a source plug of the Tape Recorder/Player subunit to the selected oPCR[2]. An example of command and response frames is shown in the table below.

Table C.2 – Example of SIGNAL SOURCE control command and response frames

field	control command frame	ACCEPTED response frame
result_status	F_{16} (default)	0_{16} (not source)
signal_source	20_{16} (Tape Recorder/Player subunit, byte 1) 00_{16} (source plug, byte 2)	<-
signal_destination	$FF7F_{16}$ (any oPCR)	$FF02_{16}$ (oPCR[2])

B. Set up a connection between unit plugs.

1 TV1 establishes a point-to-point connection between oPCR[2] of Camera1 and iPCR[1] of TV1.

C. Set up a destination device to be ready for input.

1 TV1 connects the iPCR[1] and a destination plug of the Monitor subunit by itself.

Then, the user makes Camera1 playback by a front panel key operation or by a user interface provided by TV1 that triggers sending a PLAY command to the Tape Recorder/Player subunit.

C.2 Example of a destination selection application

Two devices are involved in this application as shown in Figure C.2.

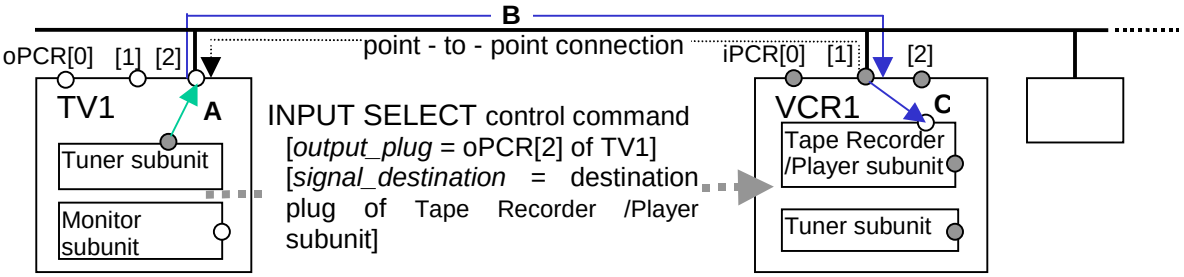


Figure C.2 – Example of a destination selection application

TV1 provides a user interface for selecting a storage device and works as a controller. A user selects VCR1 as a storage device, i.e. destination. Refer to scenario 2 also.

The following illustrates a possible procedure performed by the controller, TV1, upon a user operation.

A. Set up a source device to be ready for output.

1 TV1 connects the Tuner subunit and its oPCR[2] by itself.

B. Set up a connection between unit plugs.

1 TV1 checks the conditions of the Tape Recorder/Player subunit in VCR1 using a TRANSPORT STATE status command [R9] etc.

2 TV1 checks an internal connection in VCR1 using the SIGNAL SOURCE status command.

3 TV1 sends an INPUT SELECT control command with the CONNECT subfunction to VCR1.

The *output_plug* field for TV1 and *signal_destination* field for VCR1 are specified in the command. The *input_plug* field for VCR1 is FF_{16} (as always).

1) VCR1 determines that the appropriate plug to use is the iPCR[1] and immediately establishes a point-to-point connection between the iPCR[1] of itself and the selected oPCR[2] of TV1.

2) VCR1 sets up an internal path from the iPCR[1] to the *signal_destination* (Tape Recorder/Player subunit). After the process succeeds, VCR1 returns an ACCEPTED response to TV1. An example of command and response frames is shown in the table below.

Table C.3 – Example of INPUT SELECT control command and response frames

field	control command frame	ACCEPTED response frame
subfunction	00_{16} (CONNECT)	<-
result_status	F_{16} (default)	0_{16} (no error)
node_ID	$FFCx_{16}$ (node_ID of TV1)	<-
output_plug	02_{16} (oPCR[2] of TV1)	<-
input_plug	FF_{16} (default)	01_{16} (iPCR[1] of VCR1)
signal_destination	20_{16} (Tape Recorder/Player subunit, byte 1) 00_{16} (destination plug, byte 2)	<-

C. Set up a destination device to be ready for input.

1 Step B.3 2) above realizes this process.

C.3 Example of a preset destination selection application

One destination and one or more source devices are involved in this application. As shown in Figure C.3, TV1 works as a controller and makes preset entries on source devices. Also, Figure C.4 shows that a user operates on Camera1 as one of preset source devices to TV1. Refer to scenario 6 also.

The following illustrates a possible procedure performed by the controller, TV1, upon preset.

[PRESET process]

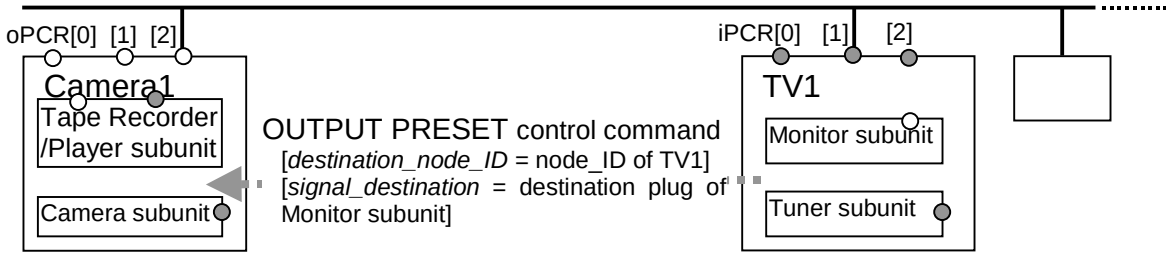


Figure C.3 – Example of a preset destination selection application (1)

1) The user selects Camera1 as a "preset source device" in the menu program of TV1.

2) TV1 sends an OUTPUT PRESET control command to Camera1.

   In the command frame, the *node_ID* of TV1 is set in the *destination_node_ID* field and a destination plug number of the Monitor subunit is set in the *signal_destination* field. In some cases, the *signal_destination* field may not be specified.

   a) Camera1 registers TV1 as the device to which it should send an INPUT SELECT control command when it requests input. An example of command and response frames is shown in the table below.

Table C.4 – Example of OUTPUT PRESET control command and response frames

field	control command frame	ACCEPTED response frame
preset_entry_number	7_{16} (default)	0_{16} (number of assigned preset entry)
destination_node_ID	FFCx_{16} (node_ID of TV1)	<-
signal_destination	00_{16} (Monitor subunit, byte 1) FF_{16} (any destination plug, byte 2)	<-

[Later, when a user starts to send a stream from Camera1 ...]

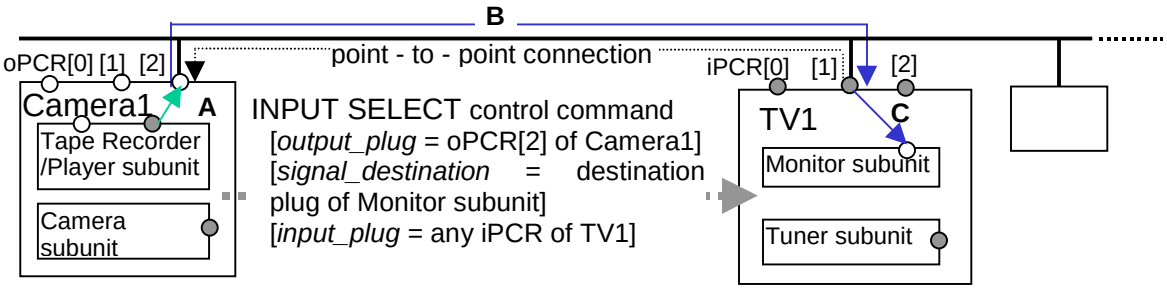


Figure C.4 – Example of a preset destination selection application (2)

The following illustrates a possible procedure performed by the controller, Camera1, upon output.

A. Set up a source device to be ready for output.

1 Camera1 sets up its internal path between its Tape Recorder/Player subunit and its oPCR[2].

B. Set up a connection between unit plugs.

1 Camera1 sends an INPUT SELECT control command with the CONNECT subfunction to TV1. An example of command and response frames is shown in the table below.

1) TV1 establishes a point-to-point connection between the oPCR[2] of Camera1 and its oPCR[1].

2) TV1 connects its iPCR[1] and a destination plug of the Monitor subunit by itself.

Table C.5 – Example of INPUT SELECT control command and response frames

field	control command frame	ACCEPTED response frame
subfunction	00_{16} (CONNECT)	<-
result_status	F_{16} (default)	0_{16} (no error)
node_ID	FFCx$_{16}$ (node_ID of Camera1)	<-
output_plug	02_{16} (oPCR[2] of Camera1)	<-
input_plug	FF_{16} (default)	01_{16} (iPCR[1] of TV1)
signal_destination	00_{16} (Monitor subunit, byte 1) FF_{16} (any destination plug, byte 2)	00_{16} (Monitor subunit, byte 1) 00_{16} (connected destination plug, byte 2)

C. Set up a destination device to be ready for input.

1 Step B.1 2) above realizes this process.

Then, TV1 selects Camera1 and displays the signal from it automatically.

C.4 Example of an input monitor application

Three devices are involved in this application. As shown in Figure C.5, TV1 provides a user interface for source device selection and works as a controller. A user selects VCR1 as a source on TV1. However, VCR1 is recording the signal from Camera1 and TV1 receives the original signal. Refer to scenario 4 also.

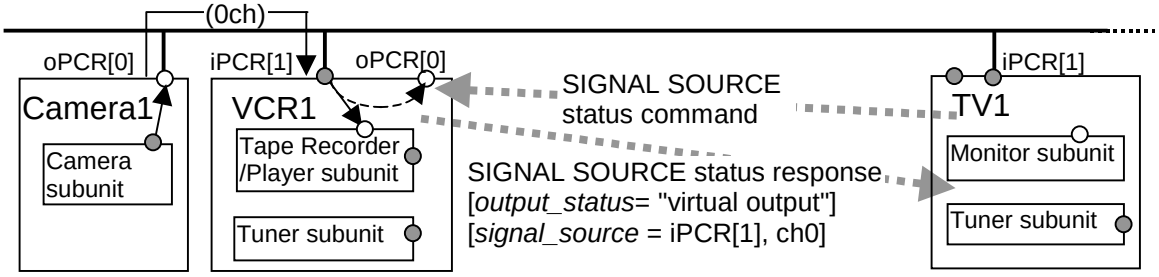


Figure C.5 – Example of an input monitor application (1)

The following illustrates a possible procedure performed by the controller, TV1, upon the selection.

A. Set up a source device to be ready for output.

1 TV1 checks an internal connection and the status of the oPCR[0] in VCR1 using a SIGNAL SOURCE status command. An example of command and response frames is shown in the table below.

1) VCR1 returns a response indicating the *output_status* of its oPCR[0] is "virtual output" and the signal source is the isochronous channel zero receiving through its iPCR[1].

Table C.6 – Example of SIGNAL SOURCE status command and response frames

field	status command frame	STABLE response frame
output_status	7_{16} (default)	4_{16} (virtual output)
conv	1_2 (default)	0_2 (can not be changed)
signal_status	F_{16} (default)	0_{16} (identical)
signal_source	$FFFE_{16}$ (default)	$C0_{16}$ (ch 0, byte 1) 01_{16} (iPCR[1], byte 2)
signal_destination	$FF00_{16}$ (oPCR[0] of VCR1)	<-

B. Set up a connection between unit plugs.

1 TV1 establishes a monitor connection to the isochronous channel zero.

C. Set up a destination device to be ready for input

1 TV1 connects its iPCR[1] and a source plug of its Monitor subunit.

TV1 displays the "virtual output" signal from VCR1. The On Screen Display of TV1 may show, for example, "VCR1 is recording". When VCR1 starts to playback, TV1 may establish a point-to-point connection between the oPCR[0] of VCR1 and its iPCR[1].

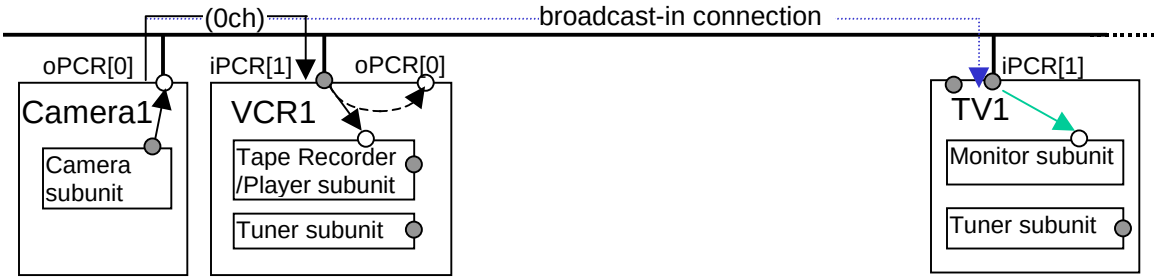


Figure C.6 – Example of an input monitor application (2)