Pin Configuration Guidelines for High Definition Audio Devices

June 17, 2013 Version 1.2

Abstract

This document provides information about Intel High Definition Audio (HD Audio) devices and drivers for the Microsoft® Windows® family of operating systems. It provides guidelines for system integrators (OEMs and ODMs) and BIOS developers for programming default values for the Pin Configuration registers in an HD Audio codec. An HD Audio codec that follows the Microsoft Universal Audio Architecture (UAA) hardware guidelines can rely entirely on the system-supplied driver support in Windows. A primary component of this driver support is the UAA HD Audio class driver, which uses the data in the Configuration Default registers to identify the individual audio devices in the codec and to make them available to the operating system and applications.

This information applies for the following operating systems:

Microsoft Windows Vista™
Microsoft Windows Server™ 2003
Microsoft Windows XP
Microsoft Windows 2000

Future versions of this preview information will be provided in the Windows Driver Kit.

The current version of this paper is maintained on the WHDC Web site at:

Pin Configuration Guidelines for High Definition Audio Devices

References and resources discussed here are listed at the end of this paper.

Document History

Date	Change
June 13, 2012	Updated flowchart in Figure 20
May 30, 2013	Deprecated redirected headphone support.

Contents

Disclaimer

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

Microsoft, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Intel High Definition Audio Disclaimer

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

CERTAIN INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This document contains information on products in the design phase of development. Do not finalize a design with this information.

Introduction

This paper provides information about Intel High Definition Audio (HD Audio) devices and drivers for the Microsoft® Windows® family of operating systems. By following the Microsoft Universal Audio Architecture (UAA) guidelines, hardware vendors and system integrators can rely entirely on the Windows operating system to provide driver support for their HD Audio codecs.

This paper presents the UAA guidelines for programming default values for the Pin Configuration registers in an HD Audio codec. The Microsoft UAA HD Audio class driver uses the information in these registers to identify the audio functions that the codec implements and to make these functions available as logical audio devices for use by the operating system and applications.

The Intel *High Definition Audio Specification* defines the requirements for an audio codec device that connects to an HD Audio link digital interface bus and is controlled by an HD Audio bus controller device. The Microsoft UAA initiative augments the Intel *High Definition Audio Specification* by providing additional hardware compatibility requirements for HD Audio codecs in Windows personal computers. For more information about the Intel HD Audio architecture and the Microsoft UAA initiative, see the References section at the end of this paper.

In Microsoft Windows Vista™, the operating system will provide a UAA HD Audio class driver. This is a system-supplied function driver for the class of devices that consists of all UAA-compliant HD Audio codecs. In addition, Microsoft is providing versions of the UAA HD Audio class driver for Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows Server™ 2003.

Hardware vendors can use the Microsoft UAA HD Audio class driver to control codec devices that conform to the Intel *High Definition Audio Specification, Revision 1.0,* and that follow the additional hardware compatibility requirements in the UAA guidelines. Hardware vendors whose audio devices meet these requirements can use the UAA HD Audio class driver to control their codecs.

A single HD Audio codec might implement several distinct audio functions. The UAA HD Audio class driver registers these functions as logical audio devices that the operating system and applications can use. For example, a particular codec might contain a device for playing audio data through a set of speakers, a device for playing audio data through headphones, and a device for recording audio data from a microphone. The UAA HD Audio class driver must be able to unambiguously determine the function that each audio device performs in an HD Audio codec, discover each control point (such as a gain control) in the device, and identify the audio jacks that are associated with the device.

The Intel *High Definition Audio Specification* defines a set of hardware mechanisms that driver software can use to identify the functions in an audio codec. The following are examples of hardware subcomponents that an HD Audio device models as a *widget*:

- A digital-to-audio converter (DAC)
- Analog-to-digital converter (ADC)
- An audio input or output connection

Widgets can be physically connected to form audio devices, and driver software can query a codec for the widgets that lie along each audio data path in the codec.

A *pin widget* is a special kind of widget that represents an audio jack or a fixed connection to an integrated speaker or microphone. Associated with each pin widget is a hardware register, called a Pin Configuration register, which provides information about the jack or fixed connection. The register contains the following information:

- Whether the pin widget is an input or an output.
- Whether the signal that passes through the pin widget is analog or digital.
- The type of external audio device (such as speakers) that connects to the pin widget.
- The type of physical connection (such as a 1/8-inch stereo jack) to the pin widget.

If the pin widget connects to an audio jack, the Pin Configuration register provides the following additional information:

- The location of the jack (such as the front panel of the system chassis).
- The color of the jack.
- Whether the jack and supporting system circuitry can detect when the user inserts or removes a plug.

Some devices, such as headphones, require a single stereo jack. Other devices might require several jacks. For example, a six-channel audio rendering device requires three stereo speaker jacks. The Pin Configuration registers inform the UAA HD Audio class driver about groups (called “associations”) of pin widgets that act together to form composite audio devices and about the roles of the individual pin widgets in each device (such as the assignment of channels in the audio stream to particular pin widgets).

Codec device manufacturers assign default values to the Pin Configuration register in each pin widget. However, the motherboard or system designer might have considerable latitude in configuring the audio devices in the codec. Thus, the default register values might not adequately describe the actual configuration of the devices in a particular system. For this reason, at power up, the system BIOS typically overwrites the default values in the Pin Configuration registers with information that more accurately describes the system-specific configurations of the audio jacks and fixed connections.

For example, a particular HD Audio codec might implement four DAC widgets that connect to four pin widgets; the pin widgets connect to either speakers or headphones. Three different motherboard or system vendors might use this collection of widgets to form the following devices:

- The first vendor might choose to use three of the pin widgets for a six-channel playback device and the fourth pin widget as an independent headphone device.
- The second vendor might choose to use the four pin widgets to implement an eight-channel playback device without headphone functionality.
- The third vendor might use three of the pin widgets as three independent stereo rendering devices and the fourth pin widget as an independent headphone device.

For each of these three systems, the system integrator (OEM or ODM) programs the system BIOS to load values into the codec's Pin Configuration registers that describe the audio devices in the system. A later section of this paper demonstrates how to program the registers to form these three sets of devices.

HD Audio codecs can vary widely in their capabilities and in the types of audio devices that they implement, but the Microsoft UAA HD Audio class driver can obtain the required information to control the devices by parsing the topologies of the devices and reading their Pin Configuration registers.

By following the guidelines in this paper for programming the default values for the Pin Configuration registers in their HD Audio codecs, vendors can ensure that the Microsoft UAA HD Audio class driver correctly identifies the audio functions in their HD Audio codecs.

UAA HD Audio Class Driver Versions

In 2006, Microsoft will release version 1.1 of the UAA HD Audio class driver. This version will be the first to provide full support for all UAA-compatible HD Audio codecs. This paper presents guidelines for designing codecs that work with versions 1.1 and later of the driver.

In 2004, Microsoft released two versions of the UAA HD Audio class driver, versions 1.0 and 1.0a, which support only a limited number of codecs. The two versions were released several months apart and provide different levels of support. Neither version provides full support for UAA-compatible HD Audio codecs.

Version 1.0 of the UAA HD Audio class driver was released in March 2004. This version does not use any Pin Configuration register information. Instead, the driver is hard-coded to support only the following small set of codecs:

- C-Media 9880
- Realtek ALC880 and ALC260

Version 1.0a of the UAA HD Audio class driver was released in October 2004. This version uses the information in the codec's Pin Configuration registers to identify the audio devices to expose. The driver lacks full support for UAA-compatible HD Audio codecs. Its support is limited to the following codecs:

- ADI 1983 and 1986
- C-Media 9880, revisions A1 and later
- Conexant Waikiki
- Realtek ALC880, revisions D and later
- Realtek ALC860, ALC260, and ALC261

Note
Versions 1.0 and 1.0a of the UAA HD Audio class driver were released before Microsoft published the guidelines in this paper. This paper describes the hardware required support for an HD Audio codec to work with versions 1.1 and later of the UAA HD Audio class driver, which is scheduled for release in late 2006. Versions 1.0 and 1.0a of the driver do not comply with these guidelines and will not be modified to make them compliant.

Microsoft currently intends to release version 1.1 of the UAA HD Audio class driver in late 2006. This version includes a built-in topology parser that provides support for UAA-compliant HD Audio codecs. This version of the driver relies on the values that it reads from the codec's Pin Configuration registers, which must comply with the Intel *High Definition Audio Specification, Revision 1.0*, and with the guidelines in this paper.

Pin Configuration Register

Each pin widget in an HD Audio codec contains a 32-bit Pin Configuration register. Driver software can query the codec for the contents of each Pin Configuration register. The register information consists of several fields, as shown in Figure 1.

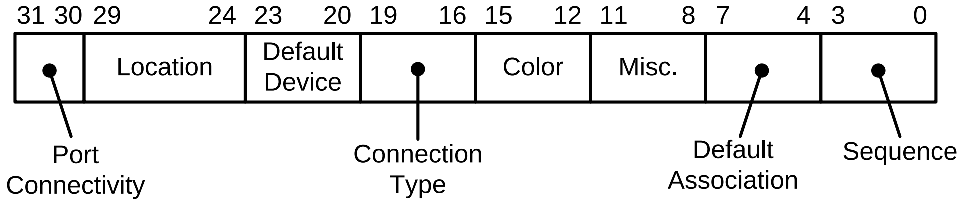


Figure 1. Pin Configuration register

For more information about the Pin Configuration register, see the Intel *High Definition Audio Specification*.

This paper presents additional, Windows-specific information about the use of Pin Configuration registers that is not in the Intel *High Definition Audio Specification*. Hardware and system vendors should follow the guidelines in this paper to ensure that they load appropriate values into these registers. The UAA HD Audio class driver requires the register values to accurately describe the way in which the codec is used in the system.

The following is a list of the fields in the Pin Configuration register and a description of how the hardware or system vendor should set each of these fields.

Port Connectivity

The Port Connectivity field indicates whether the pin widget represents an external audio jack, has a fixed connection to an internal device (such as integrated speakers), or is unconnected.

The value 0x1 indicates that the pin widget has no physical connection. The UAA guidelines for using the no-physical-connection setting, 0x1, are as follows:

- The UAA HD Audio class driver treats a pin widget with a no-physical-connection setting as a null connection.
- Use the no-physical-connection setting only when the system does not physically connect the pin widget to anything.
- For codec flexibility, generally codec hardware vendors should not set a pin widget's Port Connectivity field to 0x1 by default. Only system integrators (OEMs and ODMs) should set the field to this value (through the system BIOS).
- If the Port Connectivity field contains 0x1, never set the Default Association field to 0 because this causes the UAA HD Audio class driver to assume that the other fields in the Pin Configuration register are invalid.
- If the system integrator chooses not to use a particular pin widget, setting the Port Connectivity field for that pin to 0x1 is the *only* way to tell driver software not to use that pin widget.

There are currently no UAA-specific guidelines for Port Connectivity values other than 0x1. Use these values as appropriate to indicate the correct external connectivity. For more information, see the Intel *High Definition Audio Specification*.

Location

This field indicates the physical location of the audio jack (such as the front panel). Set the field as appropriate to indicate the location of the jack on the system or device. This information may be used to aid users, so it must be accurate. For more information, see the Intel *High Definition Audio Specification*.

Default Device

The UAA HD Audio class driver supports the following subset of the Default Device types that are defined in the Intel *High Definition Audio Specification*:

- Rendering devices: **Line Out**, **Speaker**, **HP Out**, and **SPDIF Out.**
- Capture devices: **Line In**, **Mic In**, **AUX**, and **SPDIF In.**

The UAA guidelines in this paper define additional behaviors for certain device types. For more information, see the section titled "Pin Widget Default Device Types."

Connection Type

As described previously, two or more pin widgets can be associated to form a multi-pin device. If such an association contains two or more pins of device type **Line Out** or **Speaker**, those pins must have identical values in their Connection Type fields.

If a pin widget has a Default Device field set to **Line Out**, the Microsoft UAA HD Audio class driver uses the Connection Type field to determine how to identify the single-pin or multi-pin device to which the **Line Out** pin widget belongs. If the connection type is RCA, the UAA driver identifies the device as *line connectors*. The RCA jacks of *line connectors* are suitable for driving an external device such as an audio/video receiver (AVR). For any connection type other than RCA, the **Line Out** pins in an association typically drive a set of amplified speakers and the UAA driver identifies the device as *speakers*. For associations that contain no **Line Out** pins, the connection type plays no role in device identification. For more information, see the section titled "Device Identification Algorithm."

There are currently no other UAA-specific rules for this field. Set this field as appropriate to indicate the correct connection type. For more information, see the Intel *High Definition Audio Specification*.

Color

This field indicates the color of the audio jack. The UAA HD Audio class driver uses the information from the Color and Location fields to construct user-friendly descriptions of audio jacks (such as "the orange jack on the rear panel"). The operating system and applications use these descriptions to instruct the user to plug external audio devices into the appropriate audio jacks. Providing accurate information in these fields is essential.

Each audio jack in a particular location (front panel, rear panel, and so on) must have a color that is unique to that location. These UAA guidelines ensure that the UAA HD Audio class driver can provide applications with the information that they require to unambiguously instruct users to plug their external devices into the appropriate audio jacks.

The Windows Hardware Certification recommends the color coding that is shown in Table 1.

Table 1. Audio Jack Color Coding

Color	Jack description
Pink	Microphone input
Blue	Line input
Green	Front-left and front-right speakers
Orange	Center speaker and subwoofer
Black	Back-left and back-right speakers
Gray	Side-left and side-right speakers
White	Pin connecting to analog RCA jacks
Green	Front panel headphones

For the sake of uniformity, system integrators should follow the recommended color coding where possible. However, the previously described UAA guidelines for unique colors in each association and in each physical location must be strictly observed, even when these guidelines conflict with the recommended color coding. For example, if the rear panel on a system chassis has two line-input jacks, only one of the two jacks can be blue because each jack in that location must have a unique color.

If a pin widget represents an internal connection (such as to a microphone or speakers integrated into a laptop computer), then set the Color field for the pin widget to the color code **Unknown**.

The color codes are defined in the Intel *High Definition Audio Specification*.

Miscellaneous

The Intel *High Definition Audio Specification, Revision 1.0*, defines only the Jack-Detect Override bit, which is the least-significant bit (LSB) of this 4-bit field. The Jack Detect Override bit qualifies the information in bit 2 (Presence Detect Capable) of the Pin Capabilities response format, which indicates whether the integrated circuitry for the pin widget supports jack-presence detection.

If the integrated circuitry for the pin widget supports jack-presence detection but the motherboard lacks the required external circuitry to support jack-presence detection, the BIOS should set the Jack Detect Override bit to 1 to indicate that the pin widget cannot do jack-presence detection. Otherwise, the BIOS should set this bit to 0.

The UAA guidelines for jack-presence detection are:

- A pin widget with an external connection (audio jack) and a Default Device type of **HP Out, Line In,** or **Mic In** must have jack-presence detection.
- A pin widget with an external connection and a Default Device type of **Speaker** or **Line Out** must have jack-presence detection if it plays the first two channels in an audio stream. If the stream contains more than two channels, the **Speaker** or **Line Out** pin widgets that play the remaining channels do not require jack-presence detection.

The Windows Vista logo testing will verify that HD Audio codecs meet these requirements for 3.5-mm (1/8-inch) jacks that support the above device types.

For all other types of pin widgets, see the WLP 3.0 logo requirements. At the same time, jack-presence detection is a generally useful capability for any pin widget with an external connection. The ability to expose the jack-presence detection capability of a pin widget to the operating system is being considered for version 1.1 of the UAA HD Audio class driver, regardless of whether the UAA guidelines require the pin to have it. Future versions of the UAA HD Audio class driver are likely to make increasing use of jack-presence detection to improve the dynamic behavior of audio devices.

For information about how the UAA HD Audio class driver uses jack-presence detection, see the section titled "Dynamic and Static Devices."

Default Association

This field can be used in conjunction with the Sequence field to form associations of pin widgets that operate together as multi-pin devices. All of the pin widgets in an association have the same Default Association number, but each pin widget must have a Sequence number that is unique within that association except for Associations with the number: 0xF.

A multi-pin device can use a Default Association value in the range 0x1 to 0xE. A single-pin device (an “association” that consists of a single pin widget) can use a Default Association value in the range 0x1 to 0xF. Note that Default Association value 0xF is reserved exclusively for use by single-pin devices. Although several pin widgets might use Default Association value 0xF, the UAA HD Audio class driver still treats each of the pin widgets as a single-pin device.

When examining the Default Association and Sequence fields to identify the multi-pin and single-pin devices in an HD Audio codec, the UAA HD Audio class driver behaves as follows:

- If the number of pin widgets exceeds the number of converter widgets (DACs, ADCs, and other converters), devices might have to compete for these hardware resources. Under such resource constraints, the UAA driver allocates resources that are based on priorities: when two associations require the same resource, the UAA driver assigns the resource to the association with the higher resource-allocation priority. Lower Default Association values correspond to higher priorities. Default Association value 0x1 has the highest priority, and 0xF has the lowest.
- The UAA driver enables either all or none of the widgets in an association. If the topology parser cannot find the required hardware resources to support a particular pin widget in an association, the parser rejects the entire association (and does not expose the device to the system). The UAA driver does *not* attempt to provide lesser functionality by removing the faulty pin widget from the composite device before exposing the device to the system.
- The UAA driver treats a Default Association value of 0 as invalid. If a pin widget's Default Association field is 0, the UAA driver assumes that *all* the fields in the pin widget's Pin Configuration register are invalid and ignores them. The UAA driver does not expose any device that is defined by an association that contains a pin widget with a Default Association value of 0.

Sequence

The numbers in the Sequence fields identify the individual pin widgets in an association. Valid pin sequence numbers range from 0 to 0xF. The sequence number for a pin widget in an association must be unique among all the pin widgets in that association except for association 0xF, but pin sequence numbers are not required to be unique across associations. If the association number is 0xF, every sequence number for every pin in that association must be zero.

For a single-pin device, the single sequence number in the association must always be 0.

If the same sequence number is assigned to two or more pin widgets in an association, the UAA HD Audio class driver treats the duplicate sequence number as an error. When the driver detects such an error within an association, it rejects the entire association and does not expose the device that is formed by the pin widgets in the association.

For a rendering device that plays multichannel streams, the relative position of each pin widget in a set of sequence numbers for an association implies the assignment of channels to that widget. For example, three stereo (two-channel) pin widgets must form a rendering device that plays a six-channel audio stream. If the channels are numbered 0 to 5, then the pin with the lowest sequence number plays channels 0 and 1, the pin with the highest sequence number plays channels 4 and 5, and the remaining pin widget plays channels 2 and 3.

For any composite audio device with less than 16 pin widgets, several possible sets of pin sequence numbers can be used to describe the same association. The UAA HD Audio class driver takes advantage of this redundancy to encode additional information into the sequence numbering. The UAA scheme for encoding speaker-configuration information in sequence numbers is discussed in the section titled "Speaker Configurations."

Pin Widget Default Device Types

As explained previously, the Default Device field in the pin widget's Pin Configuration register identifies the device type of the pin widget. The Intel *High Definition Audio Specification* defines the basic pin widget device types. This section presents UAA-specific information about the device types that the UAA HD Audio class driver supports.

Line Out, Speaker, and HP Out Default Device Types

The **Line Out**, **Speaker**, and **HP Out** device types are similar in function but must be carefully distinguished from each other to eliminate ambiguities. For this reason, the UAA guidelines define the following differences between these device types:

- **Line Out**. A line-out jack has no amplifier. If speakers or other external devices (such as audio/video receivers) are connected to line-out jacks, they must have their own amplifiers.
- **Speaker**. A speaker jack has an amplifier that resides on the motherboard and can be controlled by the codec (such as through the codec's external-amplifier-power-down, or EAPD, pin). The UAA HD Audio class driver turns on the on-board amplifier (EAPD) when the codec powers up and turns it off only when the codec powers down.
- **HP Out**. A headphone jack has an amplifier that is integrated into the codec and can be turned on or off by the BIOS or UAA HD Audio class driver. If the on-chip amplifier is turned off, the jack behaves like a pin widget with a Default Device type of **Line Out**; that is, the output signal passes through the pin widget without being amplified. If the on-chip amplifier is turned on, it can drive headphones but not speakers (unless they have built-in amplification). A system integrator should label a pin widget as **HP Out** only if it connects to a jack whose function is to drive headphones.

An on-*chip* amplifier, which is integrated into the circuitry of the codec device, can drive headphones but might lack sufficient power to drive unamplified speakers. An on-*board* amplifier, which resides on the motherboard and is connected to an analog output pin on the codec, should be able to drive unamplified speakers. If the motherboard contains an on-board amplifier to drive a headphone jack, the system integrator (by means of the BIOS) should set the device type of the corresponding pin widget to what it intended as default device behavior, **Speaker** or **HP Out**.

Figure 2 presents several examples of how codec vendors and system integrators should specify the Default Device types for analog output pin widgets.

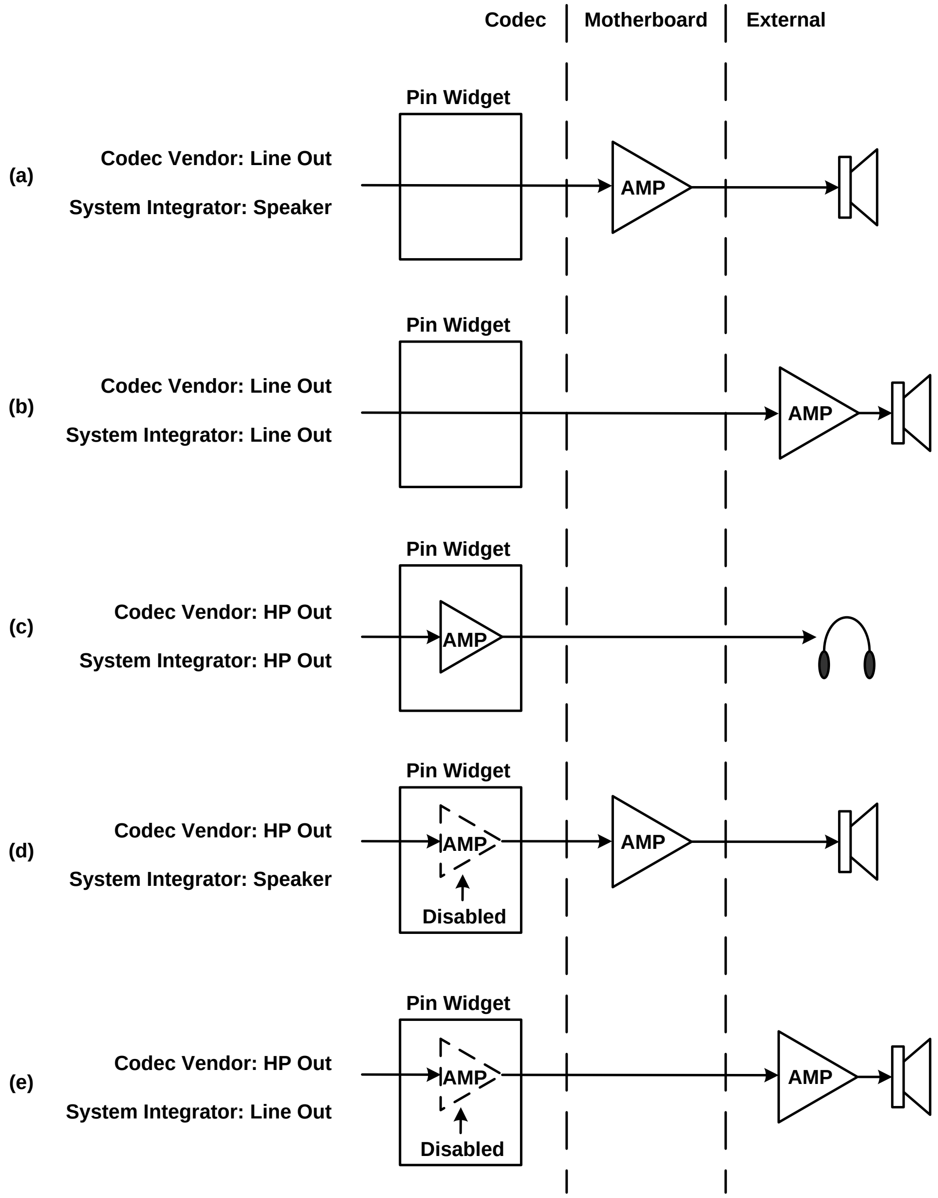


Figure 2. Default device types for analog output pin widgets

In Figure 2 (a), the pin widget has no on-chip amplifier and the codec vendor accordingly sets the pin widget's Default Device type to **Line Out**. The motherboard designer inserts an on-board amplifier between the pin widget and the jack.

Through the BIOS, the system integrator sets the Default Device type to **Speaker** to indicate the presence of the on-board amplifier.

In Figure 2 (b), the pin widget has no on-chip amplifier and the codec vendor sets the Default Device type to **Line Out**. The motherboard designer feeds the unamplified pin widget signal to the jack, and the system integrator sets the Default Device type to **Line Out**. The speakers require external amplification.

In Figure 2 (c), the codec vendor implements an on-chip amplifier in the pin widget and sets the Default Device type to **HP Out**. The motherboard designer feeds the amplified signal from the pin widget to the jack, and the system integrator sets the Default Device type to **HP Out**. Upon finding that the Default Device type is **HP Out**, the UAA HD Audio class driver turns on the on-chip amplifier.

In Figure 2 (d), the codec vendor implements an on-chip amplifier in the pin widget and sets the Default Device type to **HP Out**. The motherboard designer inserts an on-board amplifier between the pin widget and the jack. The system integrator sets the Default Device type to **Speaker** to reflect the presence of the on-board amplifier. Upon finding that the Default Device type is **Speaker**, the UAA HD Audio class driver turns off the on-chip amplifier (which will operate in pass-through mode) and turns on the on-board amplifier.

In Figure 2 (e), the codec vendor implements an on-chip amplifier in the pin widget and sets the Default Device type to **HP Out**. The motherboard designer feeds the signal from the pin widget to the jack without inserting an on-board amplifier. The system integrator sets the Default Device type to **Line Out** to indicate that the signal at the jack is unamplified. Upon finding that the Default Device type is **Line Out**, the UAA HD Audio class driver turns off the on-chip amplifier (which will operate in pass-through mode). The speakers require external amplification.

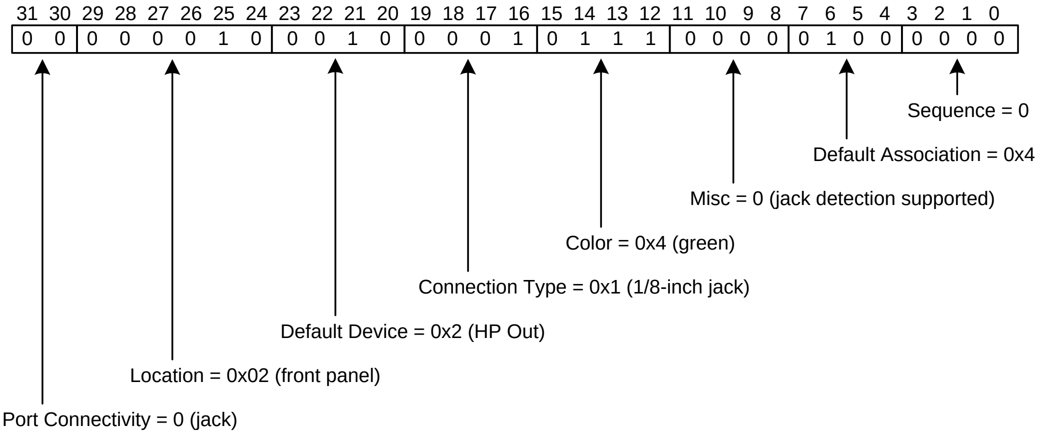
Because a **Speaker** or **Line Out** pin widget does not require an on-chip amplifier, gain and mute controls are optional for these device types. A mute control is also optional for an **HP Out** pin widget, but the UAA driver requires a gain control somewhere on the path from DAC to the HP Out pin.

If the audio path from DAC to pin for a **Speaker**, **Line Out**, or **HP Out** pin widget has a mute control, the UAA HD Audio class driver exposes the mute control to the operating system. The UAA driver also exposes the gain control in an **HP Out** pin widget to the operating system.

When defining an association of pin widgets to form a rendering device to drive a speaker configuration, the device can use either **Line Out** or **Speaker** pins to drive the speakers. However, the UAA HD Audio class driver treats any device that incorporates both **Line Out** and **Speaker** pins as an unknown (invalid) device. The two pin types are fundamentally incompatible—**Line Out** pins require external amplification, but **Speaker** pins do not.

An **HP Out** pin widget can operate either as a single-pin headphone device or as part of an association that includes one or more pin widgets of device type **Line Out** or **Speaker**. If an **HP Out** pin widget is part of an association that includes one or more **Line Out** or **Speaker** pins, the UAA driver *does* require the **HP Out** pin to support jack-presence detection so that the driver can mute the **Line Out** or **Speaker** pins if the headphones are plugged in.

Figure 3 shows the Pin Configuration register settings for an *example* headphone device with a front-panel jack.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0

Sequence = 0
Default Association = 0x4
Misc = 0 (jack detection supported)
Color = 0x4 (green)
Connection Type = 0x1 (1/8-inch jack)
Default Device = 0x2 (HP Out)
Location = 0x02 (front panel)
Port Connectivity = 0 (jack)

Figure 3. Example Pin Configuration register settings for a headphone device

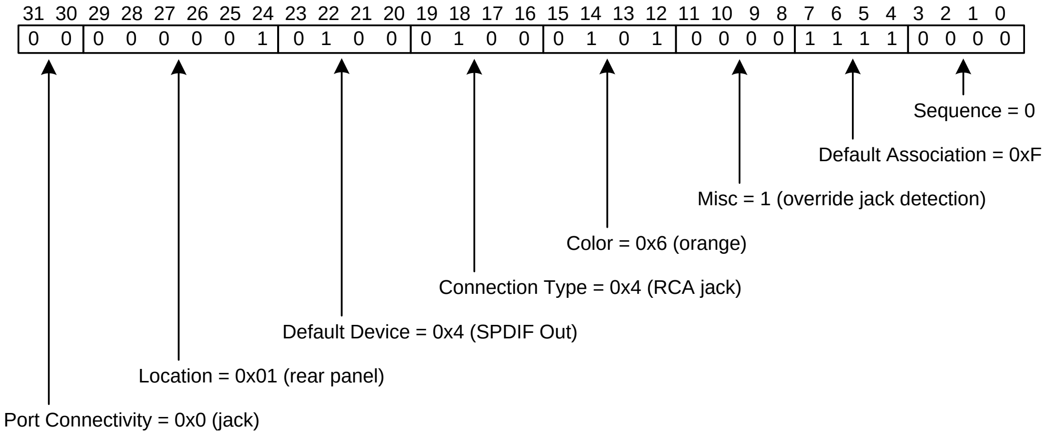
Assume that the **HP Out** pin widget in Figure 3 operates as a single-pin device, in which case it is the only pin widget in the HD Audio codec with a Default Association value of 0x4.

According to the UAA guidelines, the pin widget that connects to the headphone jack must support jack-presence detection. As shown in Figure 3, the BIOS must set the Misc field to 0 to indicate that the motherboard implements the external circuitry that is required to support jack-presence detection in addition to indicate that the integrated codec circuitry for the pin widget supports jack-presence detection.

SPDIF Out Default Device Type

A Sony/Phillips Digital Interface Format (S/PDIF) output device consists of a single pin widget of device type **SPDIF Out**. An **SPDIF Out** pin widget cannot be associated with other pin widgets of any type, including other **SPDIF Out** pin widgets. A **SPDIF Out** pin widget must have a dedicated output converter widget.

Figure 4 shows the Pin Configuration register settings for an example S/PDIF-output device with a rear-panel jack.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	0

Sequence = 0
Default Association = 0xF
Misc = 1 (override jack detection)
Color = 0x6 (orange)
Connection Type = 0x4 (RCA jack)
Default Device = 0x4 (SPDIF Out)
Location = 0x01 (rear panel)
Port Connectivity = 0x0 (jack)

Figure 4. Example Pin Configuration register settings for an S/PDIF output device

In Figure 4, the BIOS has set the Default Association field to 0xF, which is reserved for use by single-pin devices. Although the system integrator could have chosen a Default Association value in the range 0x1 to 0xE instead, this is not required because the **SPDIF Out** pin widget has a dedicated output converter and shares no hardware resources with any other pin widget.

Mic In Default Device Type

A microphone device consists of a single **Mic In** pin widget that provides either mono (one-channel) or stereo (two-channel) capture. Each **Mic In** pin widget must have a dedicated ADC, except in the special case of a mono or stereo **Mic In** pin widget that is part of an association of input pin widgets that share a single ADC through a multiplexer or mixer.

The UAA HD Audio class driver exposes a gain, boost, or mute control to the operating system if it finds a gain, boost, or mute control on the analog signal path from a **Mic In** pin widget to an ADC.

Figure 5 shows the Pin Configuration register settings for an example microphone device with a rear-panel jack and jack-presence detection.

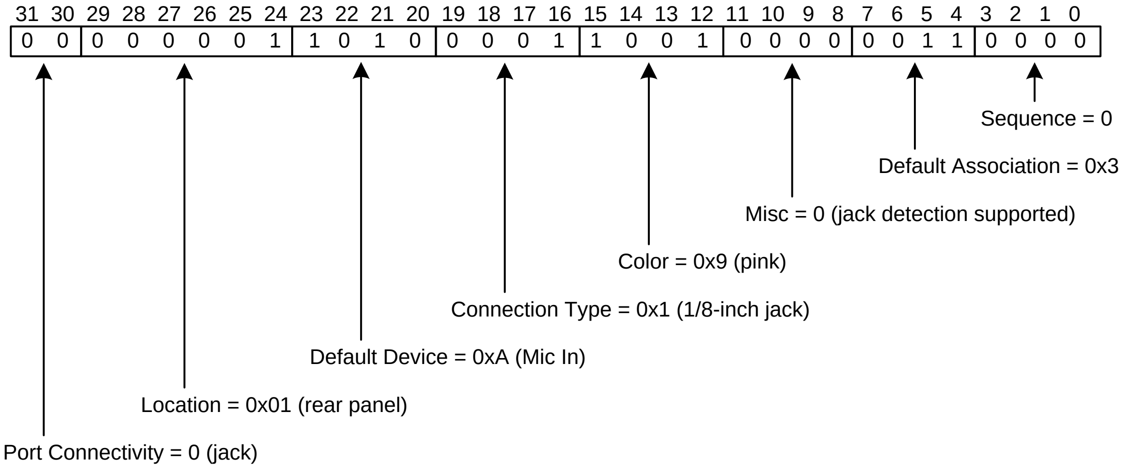

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0
Sequence = 0
Default Association = 0x3
Misc = 0 (jack detection supported)
Color = 0x9 (pink)
Connection Type = 0x1 (1/8-inch jack)
Default Device = 0xA (Mic In)
Location = 0x01 (rear panel)
Port Connectivity = 0 (jack)

Figure 5. Example Pin Configuration register settings for a microphone device

Assume that the example **Mic In** pin widget in Figure 5 operates as a single-pin device, in which case it is the only pin widget in the HD Audio codec with a Default Association value of 0x3.

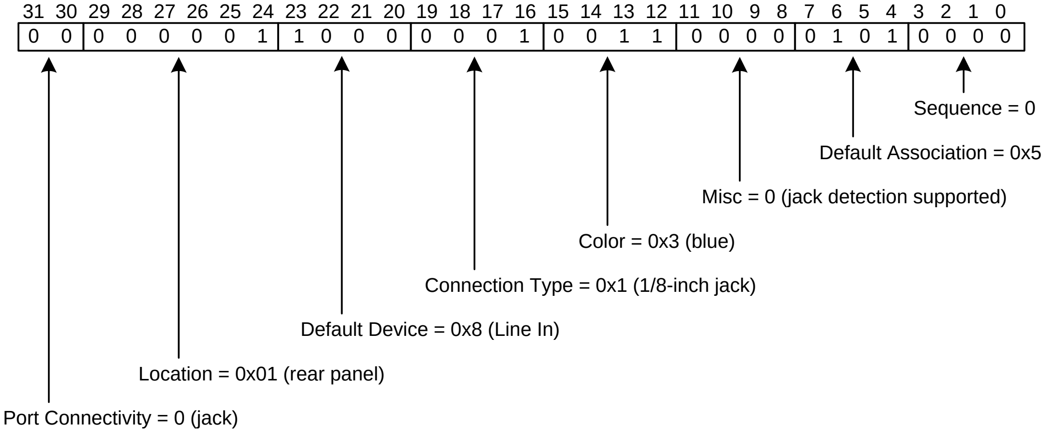
In this example, the pin widget connects to an audio jack, and the pin must therefore support jack-presence detection. In Figure 5, the BIOS has set the Misc field to 0 to indicate that the motherboard implements the required external circuitry to support jack-presence detection.

Line In Default Device Type

A line input device consists of a single **Line In** pin widget and a dedicated ADC that provides either mono (one-channel) or stereo (two-channel) capture. In addition, a mono or stereo **Line In** pin widget can be part of an association of input pin widgets that share a single ADC through a multiplexer or mixer.

The UAA HD Audio class driver does not require a **Line In** pin widget to have a gain control or mute control. However, if a **Line In** pin does have a gain or mute control, or both, the UAA driver exposes the controls to the operating system.

Figure 6 shows the Pin Configuration register settings for an example line-input device with a rear-panel jack.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0

Sequence = 0
Default Association = 0x5
Misc = 0 (jack detection supported)
Color = 0x3 (blue)
Connection Type = 0x1 (1/8-inch jack)
Default Device = 0x8 (Line In)
Location = 0x01 (rear panel)
Port Connectivity = 0 (jack)

Figure 6. Example Pin Configuration register settings for a line-input device

Assume that the example **Line In** pin widget in Figure 6 operates as a single-pin device, in which case it is the only pin widget in the HD Audio codec with a Default Association value of 0x5.

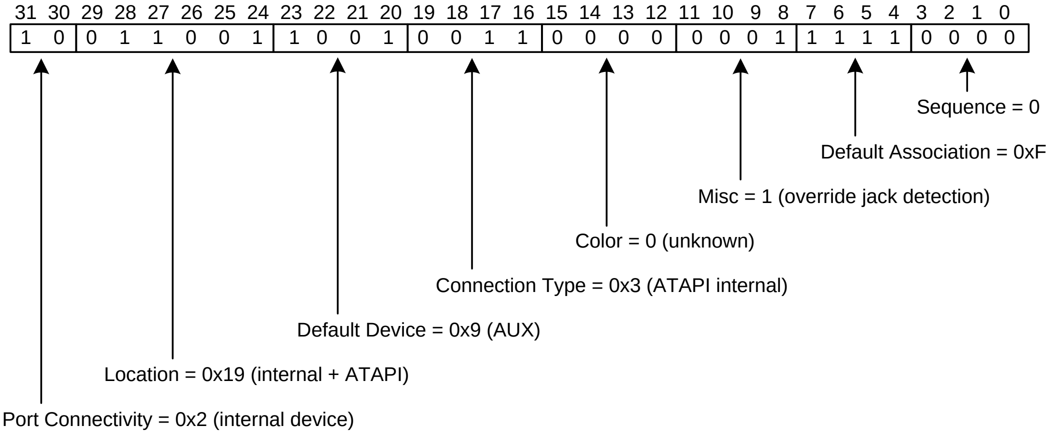
In this example, the pin widget connects to an audio jack, and the pin must therefore support jack-presence detection. In Figure 6, the BIOS has set the Misc field to 0 to indicate that the motherboard implements the required external circuitry to support jack-presence detection.

AUX Default Device Type

An auxiliary input device consists of a single **AUX** pin widget and a dedicated ADC that provides either mono (one-channel) or stereo (two-channel) capture. In addition, a mono or stereo **AUX** pin widget can be part of an association of input pin widgets that share a single ADC through a multiplexer or mixer. In either case, the connection to the **AUX** pin is internal and fixed.

The UAA HD Audio class driver does not require an **AUX** pin widget to have a gain control or mute control. However, if an **AUX** pin does have a gain control or mute control (or both), the UAA driver exposes the controls to the operating system.

Figure 7 shows the Pin Configuration register settings for an example auxiliary input device.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

Sequence = 0
Default Association = 0xF
Misc = 1 (override jack detection)
Color = 0 (unknown)
Connection Type = 0x3 (ATAPI internal)
Default Device = 0x9 (AUX)
Location = 0x19 (internal + ATAPI)
Port Connectivity = 0x2 (internal device)

Figure 7. Pin Configuration register settings for an auxiliary input device

In Figure 7, the BIOS has set the Default Association field to 0xF, which indicates that the **AUX** pin widget operates as a single-pin device. A single-pin device might require a Default Association number other than 0xF if a potential hardware resource conflict occurs, as explained previously. However, assume that in this example, the **AUX** pin widget has a dedicated DAC and shares no resources with any other pin widget.

The auxiliary input device in Figure 7 is permanently connected to an internal device. For this example, assume that the pin widget is capable of jack-presence detection (that is, the Presence Detect Capable bit of the Pin Capabilities response format is set to 1) but the motherboard does not implement the external circuitry that is required for such detection. If so, the BIOS must set the Misc field to 0x1, as shown in Figure 7, to override the Presence Detect Capable bit.

The values in the Location and Connection Type fields indicate that the pin widget is connected to an internal ATAPI device.

SPDIF In Device Type

An S/PDIF input device consists of a single pin widget of device type **SPDIF In**. A **SPDIF In** pin widget cannot be associated with other pin widgets of any type, including other **SPDIF In** pin widgets. An **SPDIF In** pin widget must have a dedicated input converter widget.

Figure 9 shows the Pin Configuration register settings for an example S/PDIF input device.

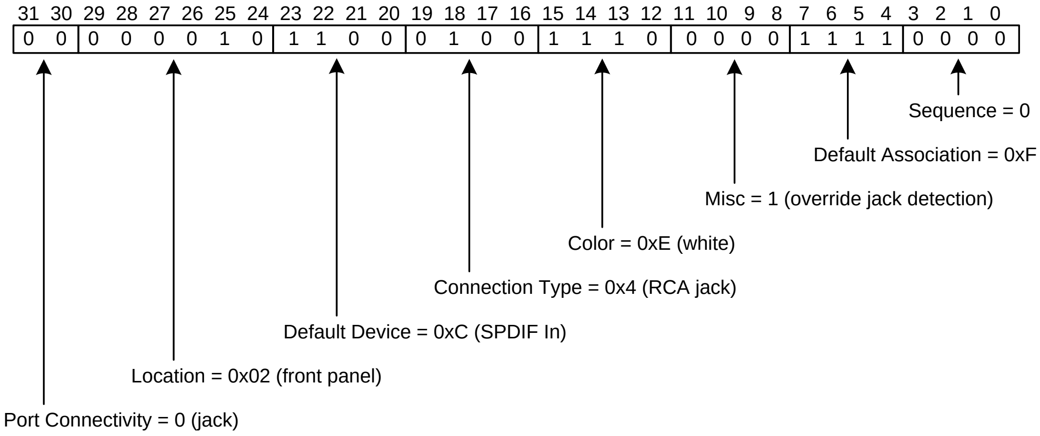


Figure 9. Example Pin Configuration register settings for an S/PDIF input device

In Figure 9, the BIOS has set the Default Association field to 0xF, which is reserved for use by single-pin devices. Although the system integrator could have chosen a Default Association value in the range 0x1 to 0xE instead, this is not required because the **SPDIF In** pin widget has a dedicated input converter and shares no hardware resources with any other pin widget.

Speaker Configurations

As explained previously, composite devices with two or more audio jacks can be formed by associating two or more pin widgets. Each pin widget in the association is assigned a sequence number and, for rendering devices, the relative position of each pin widget in the set of sequence numbers implies the assignment of channels to pin widgets.

For any composite device with less than 16 pin widgets, many possible sets of pin sequence numbers can be chosen to describe the same association.

For example, a four-channel rendering device with external speakers requires two stereo (two-channel) jacks, and any one of the following pairs of sequence numbers (shown as decimal rather than hexadecimal values) might be assigned to the pin widgets in the association:

(0, 1), (0, 2), ..., (0, 15),

(1, 2), (1, 3), ..., (1, 15),

(2, 3), (2, 4), ..., (2, 15),

.

.

.

(12, 13), (12, 14), (12, 15),

(13, 14), (13, 15),

(14, 15).

The UAA HD Audio class driver uses this redundancy to encode additional information about the rendering device into the sequence numbers. The UAA driver recognizes certain sets of sequence numbers as identifying particular speaker configurations.

For example, the sequence numbering (0, 4) identifies a quadraphonic speaker configuration in which the two channels from the first jack drive front-left and front-right speakers and the two channels from the second jack drive side-left and side-right speakers. An alternate numbering for a quadraphonic configuration is (0, 2), which again assigns the two channels from the first jack to front-left and front-right speakers but assigns the two channels from the second jack to back-left and back-right speakers. The sequence numberings (0, 2) and (0, 4) are the only sequence number pairs that the current driver recognizes as representing specific four-speaker configurations. If a four-channel rendering device uses any other pair of sequence numbers, the driver accepts the association as a valid four-channel device and exposes the device to the system, but the driver cannot determine the speaker configuration.

When the UAA HD Audio class driver recognizes a set of sequence numbers as designating a particular speaker configuration, it can provide a more meaningful name to describe the composite device. For example, it recognizes the sequence numbering (0, 2) or (0, 4) as representing a four-channel rendering device with "quadraphonic speakers." The user interface (UI) for a graphics application can use this name to better describe the device to the end user.

Table 2 shows the abbreviations for the speaker positions that this paper uses.

Table 2. Abbreviations for Speaker Positions

Abbreviation	Speaker position
FL	Front left
FR	Front right
FC	Front center
LFE	Low-frequency effects (subwoofer)
BL	Back left
BR	Back right
FLC	Front left of center
FRC	Front right of center
SL	Side left
SR	Side right

Table 3 summarizes the UAA scheme for using sequence numbers to encode speaker positions.

Table 3. Encoding of Speaker Positions

Sequence number	Left channel	Right channel
0	FL	FR
1	FC	LFE
2	BL	BR
3	FLC	FRC
4	SL	SR

In Table 3, each sequence number represents a pair of speaker positions. For example, a two-widget association with sequence numbers (0, 4) represents a four-speaker configuration with speakers located at the FL, FR, SL, and SR positions (and these speakers play channels 0, 1, 2, and 3, respectively).

The following list includes all of the speaker configurations that are recognized by the UAA HD Audio class driver, version 1.1:

- **Quadraphonic speakers**
  - Sequence numbers (0, 2) represent a four-speaker configuration with speakers that are located at the FL, FR, BL, and BR positions.
  - Sequence numbers (0, 4) represent a four-speaker configuration with speakers that are located at the FL, FR, SL, and SR positions.
- **5.1 surround sound speakers**
  - Sequence numbers (0, 1, 2) represent a six-speaker configuration with speakers that are located at the FL, FR, FC, LFE, BL, and BR positions.
  - Recommended for Windows Vista: Sequence numbers (0, 1, 4) represent a six-speaker configuration with speakers that are located at the FL, FR, FC, LFE, SL, and SR positions.
- **7.1 home theater speakers**
  - Sequence numbers (0, 1, 2, 4) represent an eight-speaker configuration with speakers that are located at the FL, FR, FC, LFE, BL, BR, SL, and SR positions.
- **7.1 wide configuration speakers** (obsolete)
  - Sequence numbers (0, 1, 2, 3) represent an eight-speaker configuration with speakers that are located at the FL, FR, FC, LFE, BL, BR, FLC, and FRC positions.
  - Sequence numbers (0, 1, 3, 4) represent an eight-speaker configuration with speakers that are located at the FL, FR, FC, LFE, FLC, FRC, SL, and SR positions.

The names for these speaker configurations are taken from the multimedia system Control Panel application, *mmsys.cpl*, in Windows XP with SP2. You can find the speaker configurations in the preceding list by following these steps:

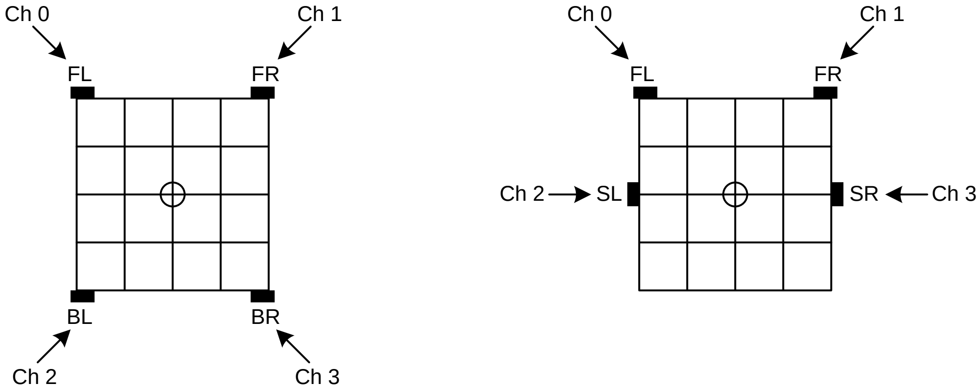
1. Open **Control Panel**.
2. In **Control Panel** (category view), click **Sounds, Speech, and Audio Devices**.
3. In the **Sounds, Speech, and Audio Devices** dialog box, click **Adjust the system volume**.
4. In the **Sounds and Audio Devices Properties** dialog box, click the **Volume** tab.
5. Under the **Speaker settings** heading, click the **Advanced** tab.
6. In the **Advanced Audio Properties** dialog box, under **Speaker setup**, open the drop-down menu and view the available speaker configurations.

The Control Panel application does not distinguish between the two quadraphonic speaker configurations, which differ only in whether BL and BR speakers are used instead of SL and SR speakers; Control Panel uses the label **Quadraphonic speakers** to identify either configuration. Similarly, it uses the label **5.1 surround sound speakers** to identify pairs of configurations that differ only in whether BL and BR speakers are used instead of SL and SR speakers. The reason for not distinguishing between the back-speaker and side-speaker configurations in either case is that home users tend not to distinguish between these speaker positions—the placement of furniture in the room might be the primary factor in determining whether a pair of speakers ends up beside or behind the listener. Forcing users to recognize the subtle differences between these alternate configurations would complicate the user interface for little benefit.

As mentioned previously, the ordering of pin sequence numbers determines the assignment of channels from the audio stream to pins: the pin with the first number in the sequence handles the first two channels, and so on. For example, the set of sequence numbers (0, 1, 2, 4) denotes a **7.1 home theater speakers** configuration that assigns the eight channels to the four stereo pins as follows:

- Channels 0 and 1 (FL and FR speakers) to the pin with sequence number 0.
- Channels 2 and 3 (FC and LFE speakers) to the pin with sequence number 1.
- Channels 4 and 5 (BL and BR speakers) to the pin with sequence number 2.
- Channels 6 and 7 (SL and SR speakers) to the pin with sequence number 4.

The following four figures show the assignment of channels to speakers for the speaker configurations that the UAA HD Audio class driver recognizes.

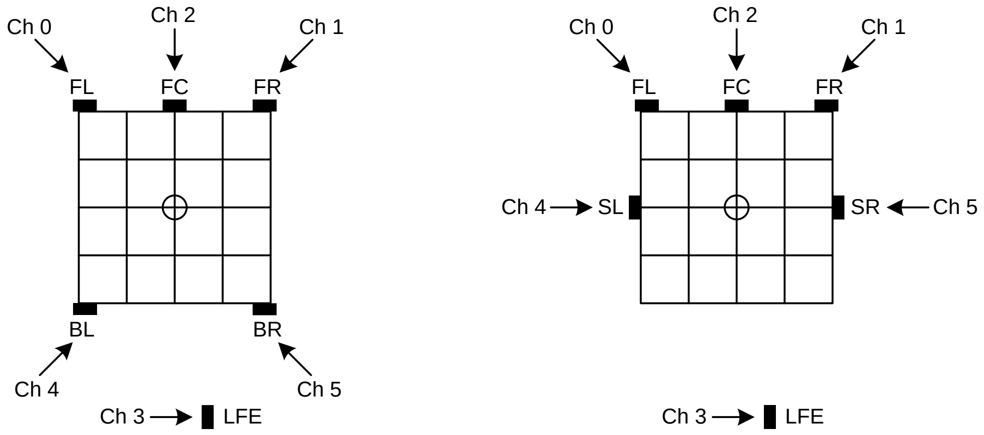


Sequence: (0, 2)

Sequence: (0, 4)

Figure 10. Quadraphonic speakers

Figure 10 shows the two speaker configurations that the multimedia system Control Panel application labels as **Quadraphonic speakers**. The left and right sides of the figure represent the speaker configurations for the sequence numberings (0, 2) and (0, 4), respectively. The circle at the center of the square grid represents the listener's position. The small black rectangles located around the perimeter of the grid represent the speakers. The arrows indicate the assignment of channels to speaker positions.

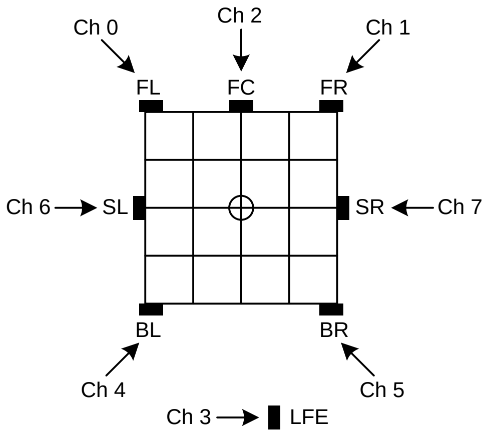


Sequence: (0, 1, 2)

Sequence: (0, 1, 4)

Figure 11. 5.1 Surround speakers

Figure 11 shows the two speaker configurations that the multimedia system Control Panel application labels as **5.1 surround speakers**. These two configurations do not assign grid positions to the LFE speakers that are based on the assumption that these speakers typically produce only low-frequency sounds, which are nondirectional.



Sequence: (0, 1, 2, 4)

Figure 12. 7.1 home theater speakers

Figure 12 shows the speaker configuration that the multimedia system Control Panel application labels as **7.1 home theater speakers**. This configuration is popular for eight-speaker home theater systems.

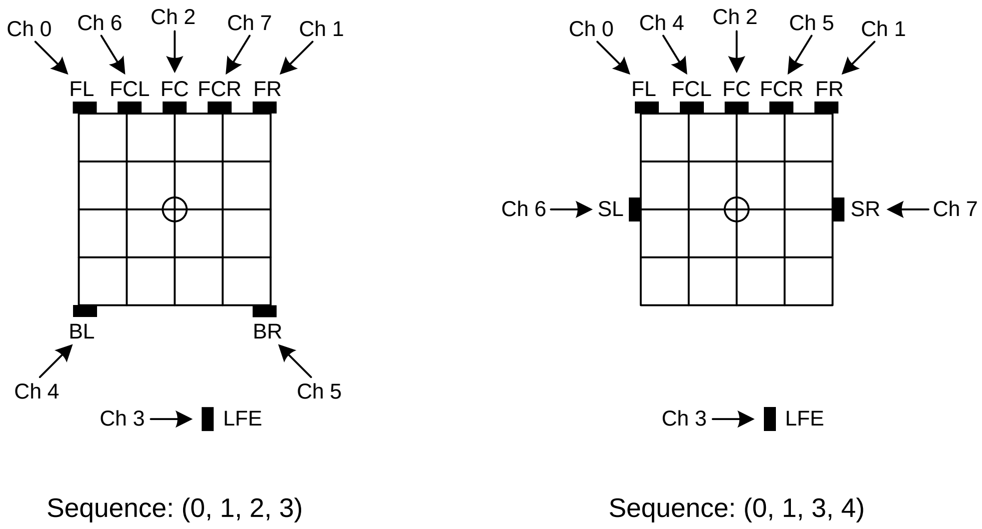


Figure 13. 7.1 wide configuration speakers

Figure 13 shows the two speaker configurations that the multimedia system Control Panel application labels as **7.1 wide configuration speakers**. These configurations are rarely used and are largely obsolete.

Hardware Resource Constraints

When registering the set of devices that are defined by the pin-widget associations in an HD Audio codec, the UAA HD Audio class driver registers only devices that have all of the hardware resources that they require to operate as independent devices. For example, if two associations include the same converter widget, the devices defined by the two associations are not independent because only one of the two devices can operate at a time due to hardware resource constraints.

If the topology parser in the UAA HD Audio class driver finds two associations that include the same widget, the parser assigns the widget to the association with the higher resource-allocation priority. As a result, the lower-priority association lacks the required hardware resources to operate as an independent device and the UAA driver registers only the higher-priority device. The UAA driver uses Default Association numbers to determine resource allocation priorities: the lower the association number, the higher the priority.

Association number 0xF, which is reserved for single-pin devices, always has the lowest priority. If several single-pin devices use association number 0xF and share the same converter widget or other hardware resource, the topology parser arbitrarily assigns the contested resource to one of the devices. Only the device that receives the resource can function as an independent device. The system integrator can avoid such unpredictable behavior by assigning a Default Association value in the range 0x1 to 0xE to the favored single-pin device and 0xF to the others.

In a few special cases, the UAA HD Audio class driver will allow pin widgets in the same association to share hardware resources. Two special cases are supported by the Windows Vista UAA HD Audio class driver:

- **Multiplexed capture device.** This audio capture device uses an input multiplexer to share an ADC between two or more analog input pins (such as

Mic In and **Line In**). The user-mode application selects an input pin on the multiplexer.

- **Mixed capture device.** This audio capture device uses an input mixer to share an ADC between two or more analog input pins. The user-mode application controls the input mix by selectively enabling and disabling the mute controls on the input pins.

For each of these two device types, the resource constraints occur between pins in the same association and not between associations. Resolving these resource conflicts is *not* the responsibility of the UAA driver. Instead, the application that uses the device is responsible for understanding the behavior that results from the internal resource constraints. For example, in the multiplexed capture device in the preceding list, an application will be unable to record audio through the microphone and line input at the same time.

If the UAA HD Audio class driver registers a multi-pin device with internal resource constraints, the driver does not register any pin widget in that device as part of another device. For example, if a **Mic In** pin widget and a **Line In** pin widget share an ADC through an input multiplexer, the system integrator can choose to have the UAA driver register only one of the following three devices:

- The special-case device that includes both the **Mic In** pin and **Line In** pin.
- A single-pin device that consists of just the **Mic In** pin.
- A single-pin device that consists of just the **Line In** pin.

Supported Devices

The UAA HD Audio class driver contains a topology parser that explores the data paths in an HD Audio codec to identify the audio devices in the codec and to discover the control points (such as gain and mute controls) in each device. The UAA driver uses the information from the topology parser to identify the valid devices in the codec.

An audio device in an HD Audio codec can consist of either a single-pin widget (and its attached converter widget) or an association of two or more pin widgets.

The UAA HD Audio class driver validates each device before registering the device to make it available to the operating system and to applications. If the UAA driver discovers a device that does not conform to the UAA guidelines and is thus invalid, the UAA driver rejects the device and the device is thereafter inaccessible to applications.

A pin-widget association must consist entirely of either input pin widgets or output pin widgets as default values. The UAA HD Audio class driver treats any device that combines input and output pin widgets as an unknown (invalid) device.

This section lists all the audio devices that the UAA HD Audio class driver recognizes as valid.

Rendering Devices

Version 1.1 of the UAA HD Audio class driver supports rendering devices that are defined as associations of the following pin widget device types:

- **Line Out**
- **Speaker**
- **HP Out**

- **SPDIF Out**

Single-Pin Rendering Devices

An association that contains a single output pin widget defines a single-pin rendering device. The pin widget in a single-pin rendering device must be one of the following device types:

- **Line Out**
- **Speaker**
- **HP Out**
- **SPDIF Out**

Multi-Pin Rendering Devices

An association of two or more output pin widgets defines a multi-pin rendering device. A multi-pin rendering device is an association of the following device types:

- Two or more **Line Out** pins
- Two or more **Speaker** pins

Combinations such as the following are considered to be redirected headphones and are legacy hardware designs:

- A single **HP Out** pin and one or more **Speaker** pins
- A single **HP Out** pin and one or more **Line Out** pins

Redirected headphones should not be implemented on new systems. These designs were suitable for Windows Vista machines, but not for any Windows 7 and later systems, since these systems include software stream redirection.

Capture Devices

Version 1.1 of the UAA HD Audio class driver supports capture devices that are defined as associations of the following pin widget device types:

- **Mic In**
- **Line In**
- **AUX**
- **SPDIF In**

Single-Pin Capture Devices

An association that contains a single input pin widget defines a single-pin capture device. The pin widget in a single-pin capture device must be one of the following device types:

- **Mic In**
- **Line In**
- **AUX**
- **SPDIF In**

Multi-Pin Capture Devices

An association of two or more input pin widgets defines a multi-pin capture device. A multi-pin capture device is an association of the following device types:

- **Mic In**
- **Line In**
- **AUX**

Version 1.1 of the UAA HD Audio class driver supports only the following two kinds of multi-pin capture device:

- **Multiplexed capture device.** An association of analog input pin widgets that share a single ADC through a multiplexer (selector).
- **Mixed capture device.** An association of analog input pin widgets that share a single ADC through a mixer (sum node).

Although sharing an ADC violates the general rule that each pin widget should have a dedicated converter, the UAA driver supports multiplexed and mixed capture devices as special cases. A multiplexed or mixed capture device can incorporate any number (including zero) of pin widgets of each of the three analog input device types: **Mic In**, **Line In,** and **AUX**.

The UAA HD Audio class driver can distinguish between a mixed capture device and multiplexed capture device simply by examining the sets of sequence numbers in the pin-widget associations that define these devices. The rules are:

- In an association that defines a mixed capture device, the BIOS must set the Sequence field in one of the pin widgets to the value 0xF.
- In an association that defines a multiplexed capture device, the BIOS must set the Sequence field in one of the pin widgets to the value 0xE.

If the sequence numbers in the association for a capture device include neither 0xE nor 0xF, the UAA driver treats the association as an unknown (invalid) device. If the sequence numbers in the association for a capture device include both 0xE and 0xF, the UAA driver treats the association as a mixed capture device.

Multiplexed Capture Device

Figure 15 shows an example of a multiplexed capture device. In this example, **Line In** and **Mic In** pin widgets share an ADC through a multiplexer (MUX).

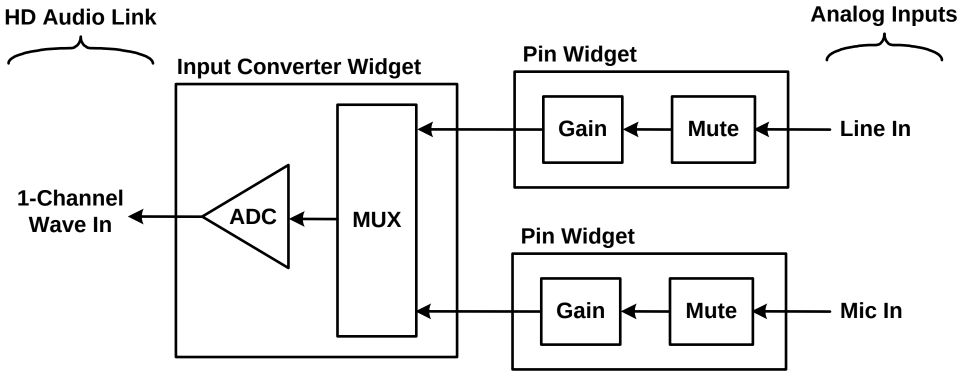


Figure 15. Two pin widgets that share a single ADC through a multiplexer

The MUX in Figure 15 selects the signal from one of the two inputs.

The example device in Figure 15 has gain and mute controls for the amplifiers in the widgets. The two pin widgets have mute controls, and all three widgets have gain controls. The mute controls are optional and can be omitted. The gain control in the **Line In** pin widget is also optional (because this type of widget does not require an on-chip amplifier), although the lack of such a control might make the device more difficult to use. However, the UAA HD Audio class driver does require a gain control either in the analog **Mic In** pin widget or in the signal path that leads from the analog **Mic In** pin widget to the ADC input. The example in Figure 15 shows gain controls in both of these locations, although one or the other would be sufficient.

Mixed Capture Device

Figure 16 shows an example of a mixed capture device. In this example, **Line In** and **Mic In** pin widgets share an ADC through a mixer (SUM).

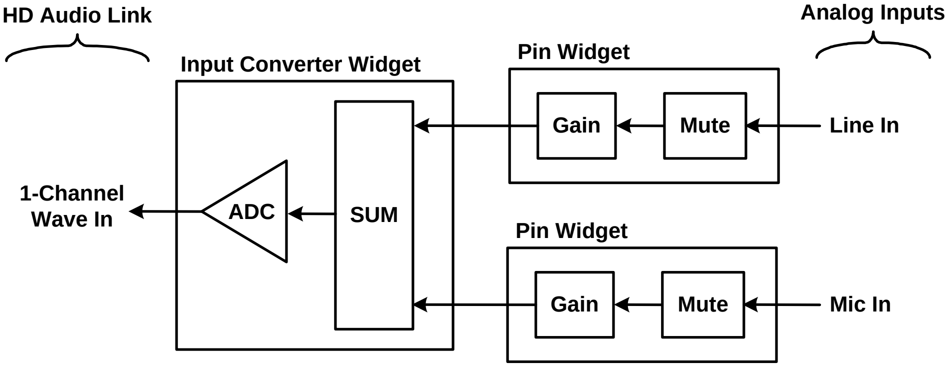


Figure 16. Two pin widgets that share a single ADC through a mixer

The mixer in Figure 16 sums the signals from the two input jacks.

The example device in Figure 16 has gain and mute controls for the amplifiers in the widgets. The two pin widgets have mute controls and all three widgets have gain controls. The mute controls are optional and can be omitted. The gain control in the **Line In** pin widget is also optional (because this type of widget does not require an on-chip amplifier), although the lack of such a control might make the device more difficult to use. However, the UAA HD Audio class driver does require a gain control either in the analog **Mic In** pin widget or in the signal path that leads from the analog **Mic In** pin widget to the ADC input. The example in Figure 16 shows gain controls in both of these locations, although one or the other would be sufficient. Gain controls in two locations on the path, such as gain and boost, can also be exposed by the Windows Vista UAA HD Audio class driver.

Device Identification Algorithm

When exploring the topology of an HD Audio codec, the topology parser in the Microsoft UAA HD Audio class driver keeps track of the widgets that it encounters along each audio data path. After the parser constructs a database of the widgets and their interconnections, the UAA driver examines the Default Association and Sequence fields in the Pin Configuration registers of all the pin widgets. The UAA driver groups together all pins that share the same Default Association values (except for the value 0xF, which is a special case, as explained previously) and attempts to identify the type of audio device that is formed by the widgets in each association.

If the UAA driver identifies a device as valid, it registers the device to make it available to the operating system and to audio applications. Device registration consists of placing a friendly name for the device and a pointer to a device interface instance in the system registry. If a device or any portion of a device is not valid, the UAA driver rejects the entire device and the ADC and DAC resources of that device are **not** made available for use by other, lower priority associations.

Line Connectors versus Speakers

Generally, the Connection Type field in the Pin Configuration register plays no role in identifying the single-pin or multi-pin device that is represented by a pin widget association.

The one exception is that the Microsoft UAA HD Audio class driver classifies an association of **Line Out** pin widgets as an audio device of type *line connectors* or *speakers*, depending on the connection type. If the **Line Out** pins connect to RCA jacks, the driver labels the device as *line connectors*. If the **Line Out** pins have any type of jack or internal connection other than RCA, the driver labels the device as *speakers*. The distinction between *line connectors* and *speakers* is primarily for the benefit of users who must select audio devices from user interfaces.

Typically, *line connectors* drive an external rendering device such as an audio/video receiver. This type of external device typically uses RCA jacks as input connectors. To preserve audio quality, the external device should receive the unamplified signals from the DACs in the codec. That is, the signals from the DAC outputs should undergo no volume-level adjustments on their way to the external device—only the external device should adjust the volume levels of the signals. The unamplified signal from a **Line Out** pin widget is suitable for use with this type of external device.

If a *speakers* device is an association of **Line Out** pin widgets, that device typically drives a set of external speakers with built-in amplifiers. If a *speakers* device is an association of **Speaker** pins, that device typically drives a set of speakers that have no built-in amplifiers. After the hardware is configured, users can usually ignore the differences between these two types of *speakers* devices.

General Control Flow

Figure 17 is a flowchart that shows the overall process that the version 1.1 UAA HD Audio class driver performs to identify the devices in an HD Audio codec.

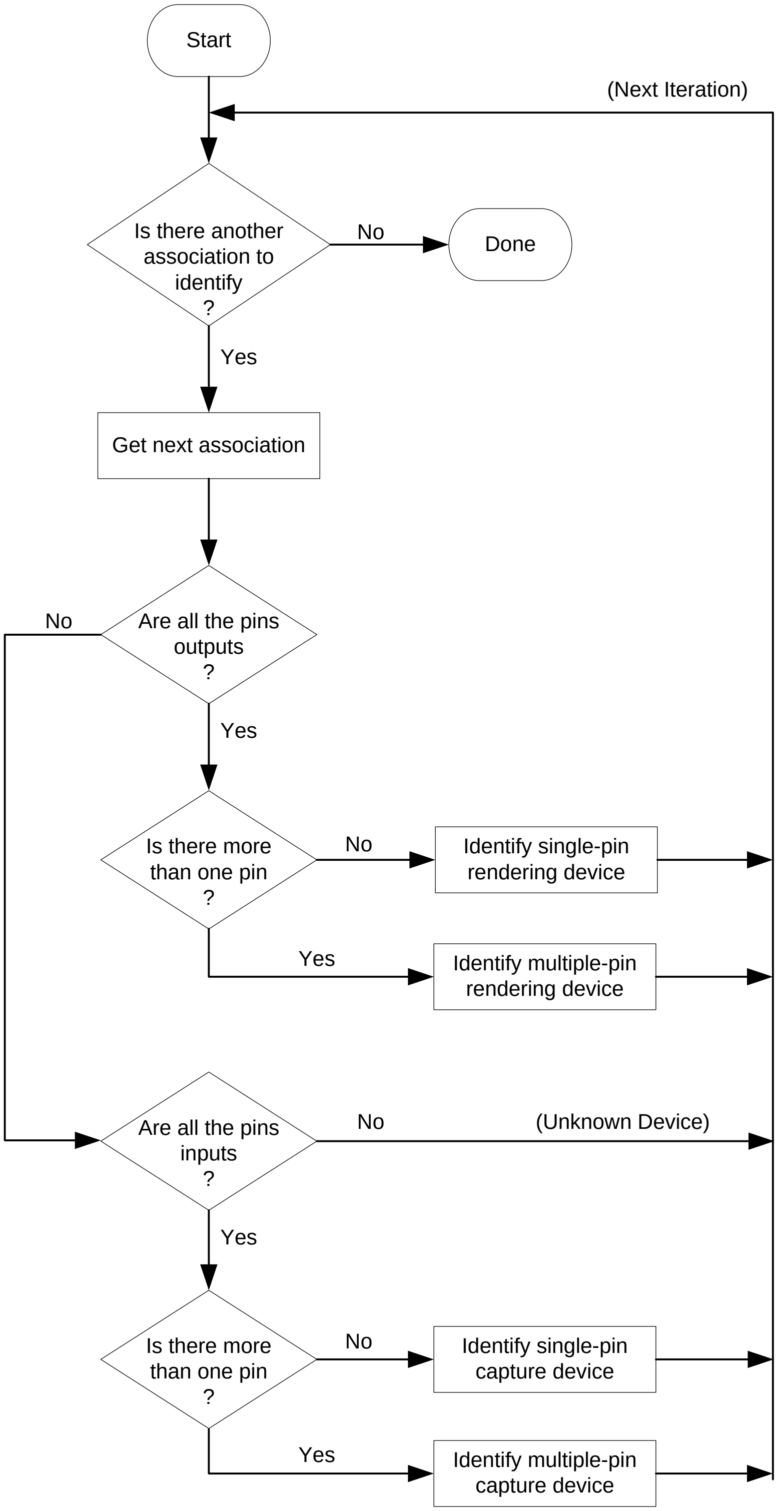

Start
(Next Iteration)
Is there another association to identify ?
No
Done
Yes
Get next association
No
Are all the pins outputs ?
Yes
Is there more than one pin ?
No
Identify single-pin rendering device
Yes
Identify multiple-pin rendering device
Are all the pins inputs ?
No
(Unknown Device)
Yes
Is there more than one pin ?
No
Identify single-pin capture device
Yes
Identify multiple-pin capture device

Figure 17. Flowchart for device-identification algorithm

Each iteration of the loop in Figure 17 processes one association of pin widgets. In Figure 17, the UAA HD Audio class driver first determines whether all the pins in the association are either inputs (for a capture device) or outputs (for a rendering device). The UAA driver recognizes the **Line In**, **AUX**, **Mic In**, and **SPDIF In** Default Device types as valid input pins. The UAA driver recognizes the **Line Out**, **Speaker**, **HP Out**, and **SPDIF Out** Default Device types as valid output pins. The UAA driver recognizes none of the additional Default Device types that are defined in the Intel *High Definition Audio Specification, Revision 1.0*. If an association contains a pin type that it does not recognize as valid, the UAA driver treats the association that contains that pin as an unknown (invalid) device.

A valid device consists entirely of either input pins or output pins. The UAA HD Audio class driver treats any device that has a combination of input and output pins as an unknown (invalid) device.

In Figure 17, the UAA HD Audio class driver's initial step in identifying a device is to determine whether it is a rendering or a capture device and whether the device consists of a single pin or multiple pins. The four boxes in the lower-right side of the figure represent four possible cases:

- Single-pin rendering device
- Multi-pin rendering device
- Single-pin capture device
- Multi-pin capture device

The following four sections describe the process for identifying each of these four general kinds of audio devices.

Single-Pin Rendering Device

The flowchart in Figure 18 shows how the version 1.1 UAA HD Audio class driver identifies a single-pin rendering device.

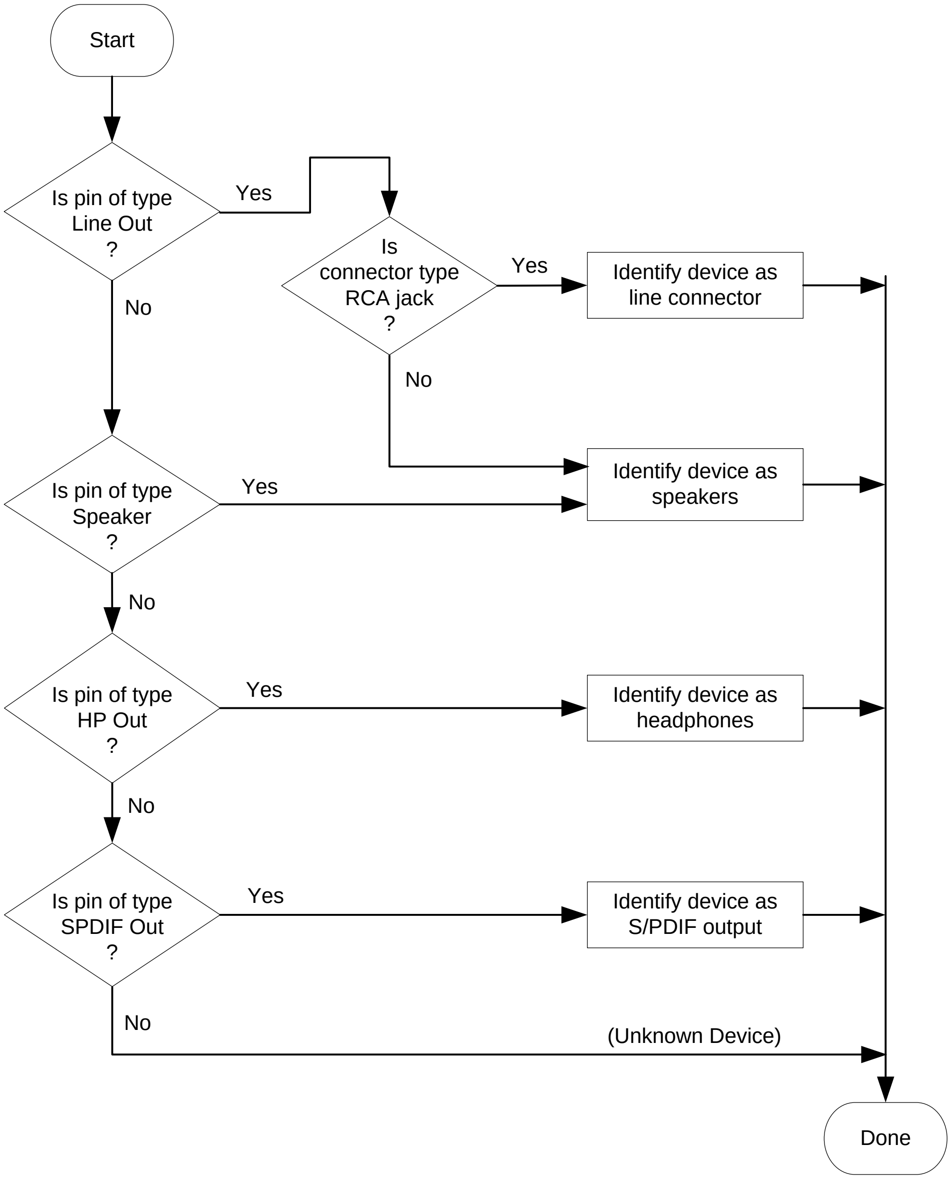

Start
Is pin of type Line Out ?
Yes
No
Is connector type RCA jack ?
Yes
No
Identify device as line connector
Is pin of type Speaker ?
Yes
No
Identify device as speakers
Is pin of type HP Out ?
Yes
No
Identify device as headphones
Is pin of type SPDIF Out ?
Yes
No
Identify device as S/PDIF output
(Unknown Device)
Done

Figure 18. Identification of single-pin rendering device

As shown in Figure 18, the UAA HD Audio class driver identifies four kinds of single-pin rendering devices: *line connectors*, *speakers*, *headphones*, and *S/PDIF outputs*.

As discussed previously, the Microsoft UAA HD Audio class driver identifies an association that consists of a single **Line Out** pin widget as either a *line connector* or *speakers*, depending on the connection type of the pin widget. Regardless of the connection type, the UAA driver identifies an association of a single **Speaker**, **HP Out**, or **SPDIF Out** pin widget as *speakers*, *headphones*, or *S/PDIF output*, respectively.

Multi-Pin Rendering Device

The flowchart in Figure 19 shows how the version 1.1 UAA HD Audio class driver identifies a multi-pin rendering device.

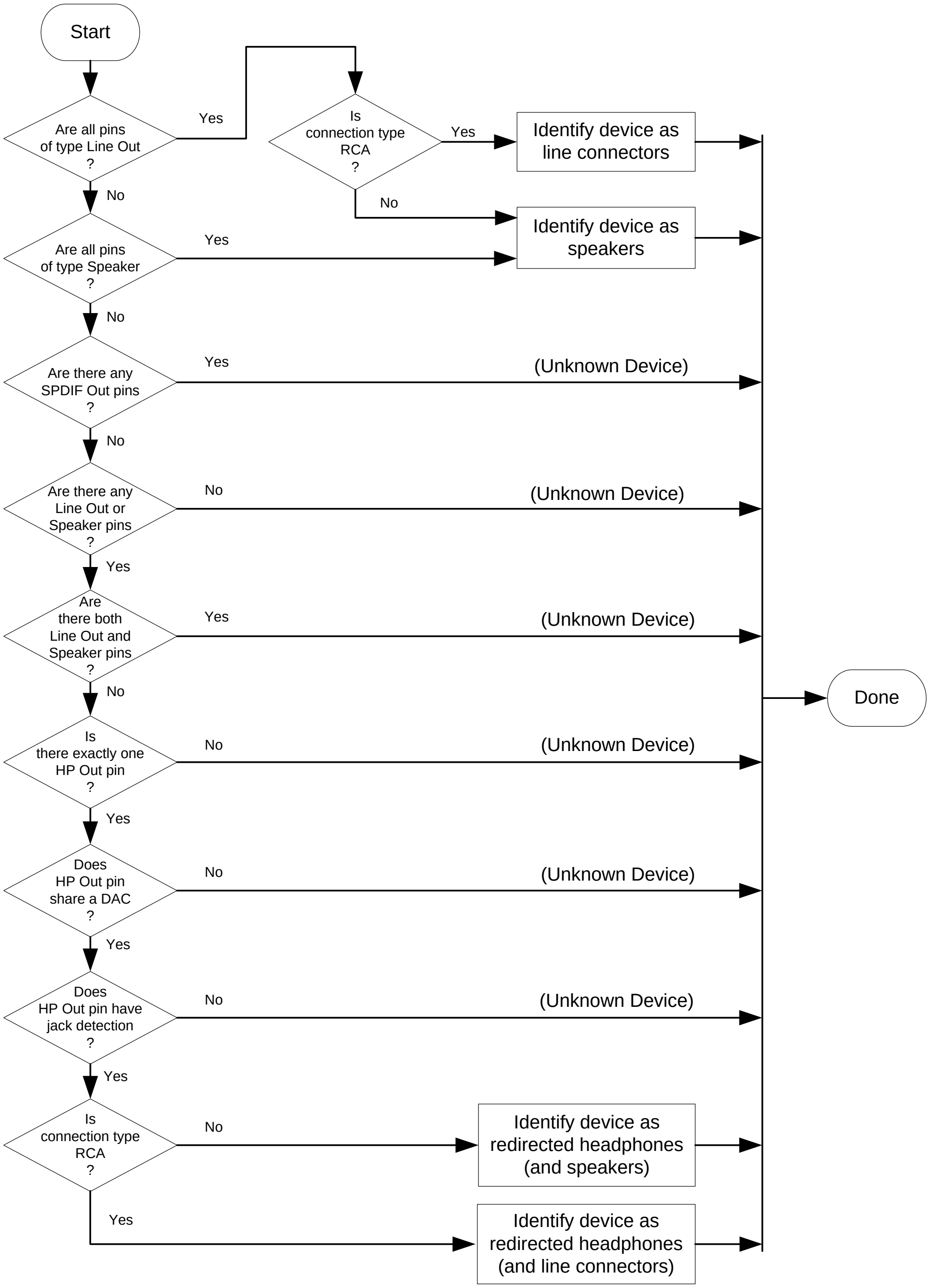

Start
Are all pins of type Line Out ?
Yes
Is connection type RCA ?
Yes
Identify device as line connectors
No
No
Identify device as speakers
Are all pins of type Speaker ?
Yes
No
Are there any SPDIF Out pins ?
Yes
(Unknown Device)
No
Are there any Line Out or Speaker pins ?
No
(Unknown Device)
Yes
Are there both Line Out and Speaker pins ?
Yes
(Unknown Device)
No
Done
Is there exactly one HP Out pin ?
No
(Unknown Device)
Yes
Does HP Out pin share a DAC ?
No
(Unknown Device)
Yes
Does HP Out pin have jack detection ?
No
(Unknown Device)
Yes
Is connection type RCA ?
No
Identify device as redirected headphones (and speakers)
Yes
Identify device as redirected headphones (and line connectors)

Figure 19. Identification of multi-pin rendering device

As shown in Figure 19, the UAA HD Audio class driver recognizes the following pin-widget associations as representing valid multi-pin rendering devices:

- All pins are of type **Line Out**.
- All pins are of type **Speaker**.

- One pin is of type **HP Out** and the remaining pins are of type **Line Out**.
- One pin is of type **HP Out** and the remaining pins are of type **Speaker**.

The last two items are redirected headphone devices, which were discussed previously.

As discussed previously, the Microsoft UAA HD Audio class driver identifies an association that consists of one or more **Line Out** pin widgets as either *line connectors* or *speakers*, depending on the connection type of the pin widgets, which must be the same for all the **Line Out** pins in the association. If the connection type is RCA, the UAA driver identifies the device as *line connectors*. For any other connection type, the UAA driver identifies the device as *speakers*.

Regardless of the connection type, the UAA driver identifies an association that consists of one or more **Speaker** pin widgets as *speakers*.

For more information about redirected headphone devices, see the section titled "Multi-Pin Rendering Devices."

Single-Pin Capture Device

The flowchart in Figure 20 shows how the version 1.1 UAA HD Audio class driver identifies a single-pin capture device.

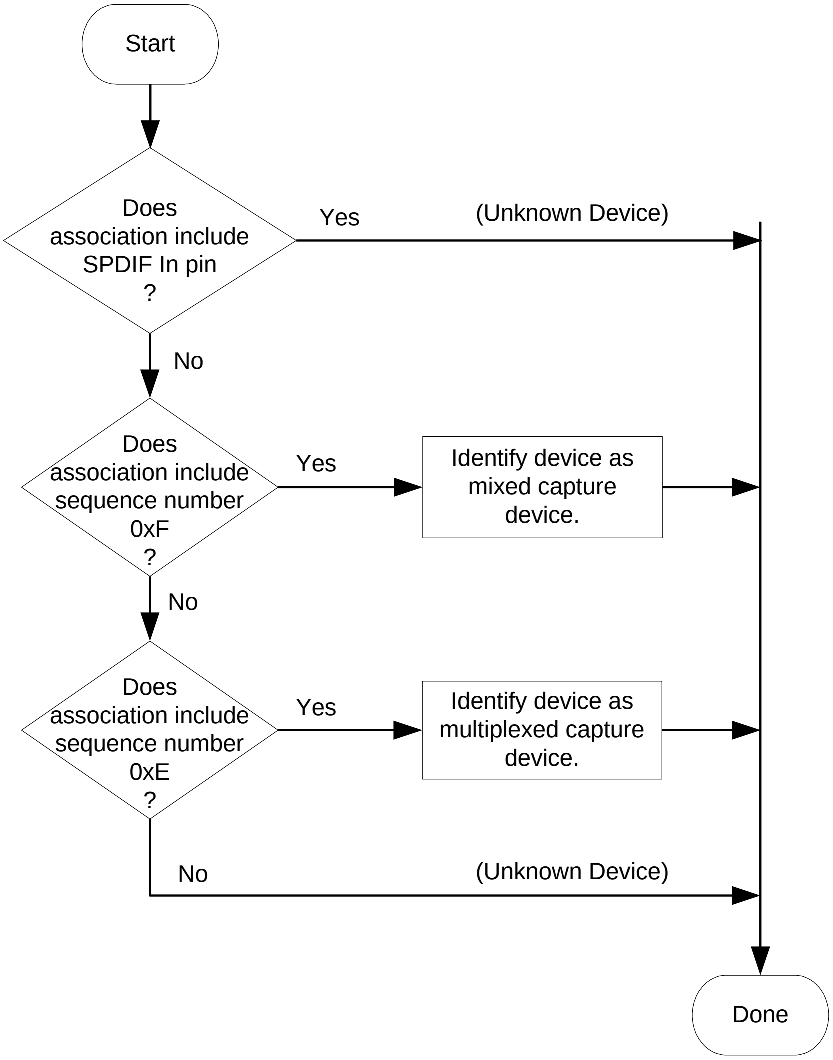

Start
Does association include SPDIF In pin ?
Yes
(Unknown Device)
No
Does association include sequence number 0xF ?
Yes
Identify device as mixed capture device.
No
Does association include sequence number 0xE ?
Yes
Identify device as multiplexed capture device.
No
(Unknown Device)
Done

Figure 20. Identification of single-pin capture device

As shown in Figure 20, the UAA HD Audio class driver identifies five kinds of single-pin capture device: *CD capture*, *line input*, *auxiliary input*, *microphone*, and *S/PDIF input*.

Multi-Pin Capture Device

The flowchart in Figure 21 shows how the version 1.1 UAA HD Audio class driver identifies a multi-pin capture device.

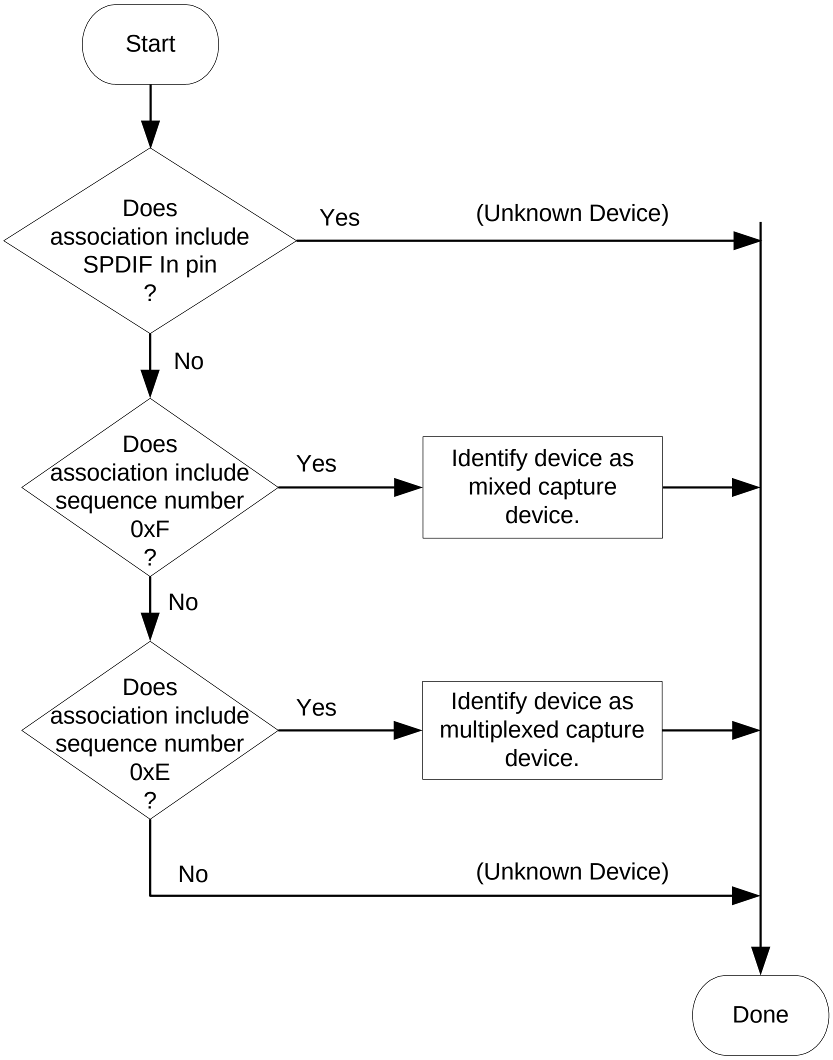

Start
Does association include SPDIF In pin ?
Yes
(Unknown Device)
No
Does association include sequence number 0xF ?
Yes
Identify device as mixed capture device.
No
Does association include sequence number 0xE ?
Yes
Identify device as multiplexed capture device.
No
(Unknown Device)
Done

Figure 21. Identification of multi-pin capture device

In Figure 21, the UAA HD Audio class driver identifies two kinds of multi-pin capture device: *mixed capture device* and *multiplexed capture device*. As explained previously, all the input pin widgets in one of these devices share a single ADC. A *mixed capture device* shares the ADC through a mixer, and a *multiplexed capture device* shares the ADC through a multiplexer. For more information, see the section titled "Multi-Pin Capture Devices."

Static and Dynamic Devices

The UAA HD Audio class driver relies on jack-presence detection to determine whether speakers, headphones, microphones, and other external devices are plugged into audio jacks. As the user plugs in and unplugs external devices, the UAA driver dynamically updates the audio devices in the system registry to reflect whether these devices are connected to external devices.

An example of a dynamic audio device is a single-pin headphone device that consists of an **HP Out** pin widget that connects to an output jack. The UAA guidelines require this type of pin widget to provide jack-presence detection. When the user unplugs the headphones, the UAA HD Audio class driver detects the change and removes the headphone device from the system registry. If the device is playing an audio stream when the plug is removed, the UAA driver stops the stream. The operating system invalidates any handles that applications might have had open on the device. Later, when the user plugs in the headphones again, the UAA driver detects the change and registers the “new” headphone device.

If a pin widget lacks jack-presence detection, the UAA HD Audio class driver must assume that an external device might be connected to the pin. In this case, the behavior of the device is static rather than dynamic. The UAA driver registers a static device following boot up, and the device typically remains in the registry until the system powers down.

If the pin widget in a single-pin device provides jack-presence detection, the UAA HD Audio class driver treats that device as a dynamic device—the device remains exposed to the use in the audio control panel only as long as the external device is plugged into the jack. If a single-pin device does not provide jack-presence detection, the UAA driver treats the device as a static device. These statements apply to all the Default Device types (analog and digital, input and output) that the UAA driver supports.

For example, a single-pin line-in device with jack-presence detection is a dynamic device, and a single-pin line-in device without jack-presence detection is a static device.

In the case of a multi-pin device, the UAA HD Audio class driver either registers the entire device—consisting of all the pin widgets in the association—or it does not register the device at all. The UAA driver never attempts to register a subset of the device that consists of the pin widgets that are currently connected to external devices.

The following sections describe the dynamic behavior of multi-pin devices.

Dynamic Multi-Pin Rendering Devices

As mentioned previously, the UAA guidelines require jack-presence detection in the **Speaker** or **Line Out** pin widget that connects to the first speaker jack—the jack that plays the first two channels of an audio stream. The **Speaker** or **Line Out** pin widgets that play the additional channels in a multichannel stream can, as an option, provide jack-presence detection as well, but the UAA HD Audio class driver does not use the jack-presence detection information from these other pins to manage the dynamic behavior of the device.

In the case of a multi-pin rendering device that consists entirely of **Speaker** or **Line Out** pin widgets, the UAA HD Audio class driver registers the device only if the first speaker jack is plugged in. When the user unplugs the first speaker jack, the UAA driver detects the change and removes the device from the audio user interface. When the user plugs the first speaker jack in again, the UAA driver detects the change and registers the device again.

In a redirected headphone device, the behavior is similar to that just described for the multi-pin rendering device with all **Speaker** or **Line Out** pins. However, the UAA HD Audio class driver removes a redirected headphone device from the user interface only if the first speaker jack and headphones (**HP Out** jack) are both unplugged. When the user plugs either the speakers or headphones in again, the UAA driver detects the change and registers the device again.

The **Speaker** or **Line Out** pin widgets in a redirected headphone device can be internally connected to speakers that are, for example, integrated into the chassis of a laptop computer. In this case, the pin widget for the first two speaker channels does not provide jack-presence detection. Thus, the device behavior is static rather than dynamic.

Dynamic Multi-Pin Capture Devices

A multiplexed capture device or mixed capture device can be a combination of pin widgets with and without jack-presence detection. The UAA HD Audio class driver registers this type of device in either of the following situations:

- All the pin widgets in the device have jack-presence detection but at least one of the jacks is plugged in.
- One or more pin widgets in the device lack jack-presence detection.

If all of the inputs to a multiplexed or mixed capture device provide jack-presence detection (a Windows Vista Logo requirement), then the UAA HD Audio class driver maintains the device in the registry (exposed to the user and the system) as long as any one of the inputs is plugged in. However, if the user unplugs all the jacks, the UAA driver responds to the unplugging of the last jack by removing the device from the registry (and from the user/system). If the user later plugs in any single jack, the UAA driver registers the device again.

If any pin widget in a multiplexed or mixed capture device lacks jack-presence detection, the UAA HD Audio class driver must assume that the pin widget might be connected to an external device. In this case, the device registration is static—the UAA driver registers the multiplexed or mixed capture device following boot up, and the device typically remains registered until the system powers down.

Pin Configuration Examples

This section presents the details of the three Pin Configuration register programming examples that were mentioned in the introduction.

All three examples use the same set of four DAC widgets from an HD Audio codec, but each example configures the same four widgets in a different way to form a different set of audio rendering devices. Each DAC widget connects to a pin widget that represents a stereo output jack. The motherboard or system vendor configures the pin widgets by choosing appropriate settings for the Pin Configuration registers that belong to the pin widgets.

The HD Audio codec contains four DAC widgets that are identical to the one shown in Figure 22.

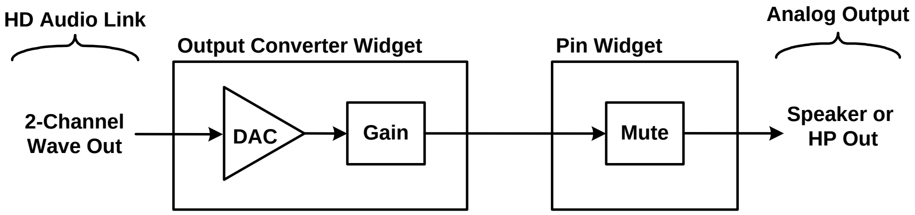


Figure 22. A single-pin analog rendering device

In the example in Figure 22, the DAC widget has a gain control and connects to a pin widget that has a mute control.

Three different motherboard or system vendors can use the same set of four widgets to form the following devices:

- *Example 1.* The first vendor can choose to use three of the pin widgets to form a six-channel rendering device (such as a **5.1 surround speakers** configuration, as described in the previous section titled "Speaker Configurations") and use the fourth pin widget for an independent headphone device.
- *Example 2.* The second vendor can choose to use the four pin widgets to implement an eight8-channel rendering device (such as a **7.1 home theater speakers** configuration) without headphone functionality.
- *Example 3.* The third vendor can use three of the pin widgets as three independent two-channel rendering devices (such as stereo speakers) and use the fourth pin widget as an independent headphone device.

For each of these three systems, the system integrator (OEM or ODM) programs the system BIOS to load values into the codec's Pin Configuration registers that describe the audio devices in the system. The details of these three examples are presented in the following sections.

Example 1

The first vendor configures the four pin widgets by loading the settings shown in Table 4 into the four Pin Configuration registers that belong to the pin widgets.

Table 4. Pin Configuration Register Settings for Example 1

Register field	Pin A	Pin B	Pin C	Pin D
Port Connectivity	0x0 (jack)	0x0 (jack)	0x0 (jack)	0x0 (jack)
Geometry Location	0x1 (rear)	0x1 (rear)	0x1 (rear)	0x2 (front)
General (Gross) Location	0x0 External on primary chassis	0x0 External on primary chassis	0x0 External on primary chassis	0x0 External on primary chassis
Default Device	0x0 (**Line Out**)	0x0 (**Line Out**)	0x0 (**Line Out**)	0x2 (**HP Out**)
Connection Type	0x1 (1/8-inch)	0x1 (1/8-inch)	0x1 (1/8-inch)	0x1 (1/8-inch)
Color	0x4 (green)	0x6 (orange)	0x2 (gray)	0x7 (green)
Misc	0 (jack detect)	0 (jack detect)	0 (jack detect)	0 (jack detect)
Default Association	0x5	0x5	0x5	0xF
Sequence	0 (FL, FR)	0x1 (FC, LFE)	0x4 (SL, SR)	0

Pins A, B, and C drive a set of six speakers, and pin D drives a set of stereo headphones.

Pins A through C belong to the same pin-widget association. Hence, they operate together as a single device. The topology parser recognizes the sequence numbers (0, 1, 4) for pins A through C as identifying a **5.1 surround sound speakers** configuration, as described in the previous section titled "Speaker Configurations."

The Default Association number for the headphones device is 0xF. By convention, the UAA HD Audio class driver treats all pin widgets with Default Association number 0xF as single-pin devices. Although these pin widgets have the lowest priority in hardware resource conflicts, pin D in this example has a dedicated DAC widget and no resource conflicts with any other pin widgets. Hence, a low priority cannot possibly cause pin D to lose any hardware resources. However, if pin D *did* have a resource conflict, assigning the pin a higher priority (that is, a Default Association number less than 0xF) would be necessary to ensure that the UAA driver assigns the pin the resources that it requires to function as a device.

In accordance with the previously described color-coding recommendations, the jack colors for pins A, B, and C are as follows:

- Pin A, which transmits channels 0-1 (FL and FR speakers), is green.
- Pin B, which transmits channels 2-3 (FC and LFE speakers), is orange.
- Pin C, which transmits channels 4-5 (SL and SR speakers), is gray.

Example 2

The second vendor configures the four pin widgets by loading (using system BIOS) the settings shown in Table 5 into the four Pin Configuration registers that belong to the pin widgets.

Table 5. Pin Configuration Register Settings for Example 2

Register field	Pin A	Pin B	Pin C	Pin D
Port Connectivity	0 (jack)	0 (jack)	0 (jack)	0 (jack)
Geometry Location	0x01 (rear)	0x01 (rear)	0x01 (rear)	0x01 (rear)
General (Gross) Location	0x0 External on primary chassis	0x0 External on primary chassis	0x0 External on primary chassis	0x0 External on primary chassis
Default Device	0x0 (**Line Out**)	0x0 (**Line Out**)	0x0 (**Line Out**)	0x0 (**Line Out**)
Connection Type	0x1 (1/8-inch)	0x1 (1/8-inch)	0x1 (1/8-inch)	0x1 (1/8-inch)
Color	0x4 (green)	0x6 (orange)	0x1 (black)	0x2 (gray)
Misc	0 (jack detect)	0 (jack detect)	0 (jack detect)	0 (jack detect)
Default Association	0x2	0x2	0x2	0x2
Sequence	0 (FL, FR)	0x1 (FC, LFE)	0x2 (BL, BR)	0x4 (SL, SR)

Pins A, B, C, and D comprise a multichannel rendering device that drives a set of eight speakers. The topology parser recognizes the sequence numbers (0, 1, 2, 4) for pins A through D as identifying a **7.1 home theater speakers** configuration, as described in the previous section titled "Speaker Configurations."

In accordance with the previously described color-coding recommendations, the jack colors for pins A through D are as follows:

- Pin A, which transmits channels 0 and 1 (FL and FR speakers), is green.
- Pin B, which transmits channels 2 and 3 (FC and LFE speakers), is orange.
- Pin C, which transmits channels 4 and 5 (BL and BR speakers), is black.
- Pin D, which transmits channels 6 and 7 (SL and SR speakers), is gray.

Example 3

The third vendor configures the four widgets by loading the settings shown in Table 6 into the four Pin Configuration registers that belong to the pin widgets.

Table 6. Pin Configuration Register Settings for Example 3

Register field	Pin A	Pin B	Pin C	Pin D
Port Connectivity	0x2 (fixed)	0 (jack)	0 (jack)	0 (jack)
Geometry Location	0x0 (not available)	0x01 (rear)	0x01 (rear)	0x02 (front)
General (Gross) Location	0x1 Internal	0x0 External on primary chassis	0x0 External on primary chassis	0x0 External on primary chassis
Default Device	0x1 (**Speaker**)	0x0 (**Line Out**)	0x0 (**Line Out**)	0x2 (**HP Out**)
Connection Type	0x7 (other analog)	0x1 (1/8-inch)	0x1 (1/8-inch)	0x1 (1/8-inch)
Color	0x0 (unknown)	0x4 (green)	0x4 (green)	0x4 (green)
Misc	0x1 (override)	0 (jack detect)	0 (jack detect)	0 (jack detect)
Default Association	0x1	0x2	0x3	0x4
Sequence	0	0	0	0

Pins A, B, and C each drive a set of stereo speakers and pin D drives a set of stereo headphones.

The Default Association number for pins B through D is 0xF, which means that each operates as a single-pin device. In addition, each of pins B through D has dedicated hardware resources that it does not share with other pins. Hence, none of these pins needs a Default Association number lower than 0xF.

In contrast, pin A in this example shares a hardware resource, such as a DAC widget, with some other pin widget (not shown in Table 6). To ensure that the UAA HD Audio class driver assigns the contested resource to pin A rather than to the other pin widget, the BIOS sets the Default Association number for pin A to 0x7 and the Default Association number for the other pin widget to 0xF. Note that this is a type of headphone redirection. These designs were suitable for Windows Vista machines but not for any Windows systems which include software stream redirection (Windows 7 and later)

Another way that pin A differs from pins B through D is that pin A has a fixed, internal connection to a set of stereo speakers that are integrated into the system chassis. In contrast, the other three pins have audio jacks that connect to external speakers (pins B and C) or headphones (pin D).

Conclusion

Motherboard and system vendors can rely entirely on the Windows operating system to provide the driver support for their HD Audio solutions. HD Audio codecs are flexible hardware devices that can be configured by system BIOS routines to implement a variety of logical audio devices to support the designs of individual systems. In each system, the BIOS loads the Pin Configuration registers in an HD Audio codec with default settings that identify the audio devices that the codec implements in the system. This paper has presented a set of guidelines for programming the registers. By following these guidelines, motherboard and system vendors can ensure that the Microsoft UAA HD Audio class driver correctly identifies each audio device and makes that device available to Windows applications.

Call to Action

System integrators (OEMs and ODMs) and BIOS developers:
Follow the guidelines in this document to program appropriate default values for the Pin Configuration registers in the HD Audio codec with the information that the UAA HD Audio class driver requires to accurately identify the audio devices in your system.

Use Microsoft HD Audio Pin Configuration Validation Tools that are available from uaa@microsoft.com to ensure logo compliance of your Windows Vista system design.

Use the Microsoft UAA HD Audio class driver as part of your system bring-up test matrix. The drivers ships in Windows Vista builds but can also be obtained in sometimes more recent version from uaa@microsoft.com.

Codec hardware manufacturers:
Follow the guidelines in this document to program appropriate default values for the Pin Configuration registers in your HD Audio codec. Educate your customers about how to configure the audio devices in their Windows systems—by following the guidelines, they can make more of the capabilities of those devices available to users.

Hardware vendors:
Propose new device types to Microsoft if you see a strong customer requirement for them. Microsoft will consider supporting new device types in future versions of the class driver.

Microsoft is working with hardware vendors and system integrators to ensure that the UAA HD Audio class driver provides full support for the audio devices in their HD Audio codecs. To ensure complete driver support for your codec, contact Microsoft.

If you have questions about UAA, send an e-mail to uaa@microsoft.com.

References

- For information about Windows compatibility guidelines for UAA devices, including HD Audio solutions, review the Windows Logo Program for Hardware requirements at http://www.microsoft.com/whdc/winlogo/hwrequirements.mspx.
- For information about the Intel HD Audio architecture, see the Intel *High Definition Audio Specification* at http://www.intel.com/standards/hdaudio/.
- For an overview of the Microsoft UAA initiative, see the white paper titled *Universal Audio Architecture* at http://www.microsoft.com/whdc/device/audio.
- For information about providing Plug and Play support for HD Audio devices, see the white paper titled *Plug and Play Guidelines for High Definition Audio Devices* at http://www.microsoft.com/whdc/device/audio/HD-aud_PnP.mspx.

Other Resources

WHDC

Technical information, development and testing kits, and other essential resources for system and device manufacturers, driver developers, and other professionals who create products that run with Windows.
http://www.microsoft.com/whdc/default.mspx