Qualcomm Technologies International, Ltd.

ADK 4.3 Audio Source Application

User Guide

80-CF409-1 Rev. AA

November 9, 2017

Qualcomm Technologies International, Ltd. (formerly known as Cambridge Silicon Radio Limited) is a company registered in England and Wales with a registered office at: Churchill House, Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ, United Kingdom. Registered Number: 3665875 | VAT number: GB787433096

Revision history

Revision	Date	Description
AA	November 2017	Original publication of this document. Alternative document number CS-00406741-UG. Updated for ADK 4.3 and based on CS-00224993-UG.

Contents

Tables

Figures

1 ADK Audio Source application - overview

The Audio Source application runs on QTIL CSR8670 and CSR8675 ICs. It is connected, using Bluetooth, to a Mono or Stereo Audio Sink, for example to a:

- Headset
- Soundbar
- Speaker

It enables a Bluetooth device to make/receive Voice over IP (VoIP) calls and to listen to music playing on a Personal Computer (PC). It also enables other portable music player devices, such as media players like iPods or mobile phones, to listen to music playing over Bluetooth.

The Audio Source application is a plug-and-play solution. Therefore, there is no need to install any software on the PC or other music playing devices.

The Audio Source application works in four modes:

- USB
- Analog
- SPDIF
- I^2S

In USB mode, when plugged in, the audio source enumerates as a soundcard and becomes the default audio device on the PC. It is compatible with all major VoIP software and most popular music providers.

In Analog mode, the audio output from the music player is fed in to the Audio Source application. As a result, it is suitable with portable media players and PCs.

In SPDIF mode, the source application accepts a PCM SPDIF input and streams the audio over a Bluetooth link to an audio sink.

In the I^2S mode, the source application accepts I^2S input from an ATF board and streams the audio over Bluetooth link to an audio link.

The Audio Source application is customized using the configuration PS Keys described in PS Key configuration in the ADK Audio Source application. This enables the behavior of the audio source to be altered without changing any source code.

2 To flash the Audio Source application onto suitable hardware

1. Open **xIDE** from the ADK program folder.
2. Select **Project > Open Workspace** and choose the source application from the **apps** subdirectory.
3. The application is set up to run on CSR8670/CSR8675 hardware in USB mode. To change hardware or audio mode, see To set the xIDE project environment.
4. . Select **Debug > Transport** and choose the required Transport.
5. Select **Build > Build Active Project** (**F7**).
6. Select **Debug > Run** (**F5**).
7. The relevant PS Keys are automatically merged onto the hardware by the Source project.

   NOTE Merge the PS Keys manually using the **PSTool** application if no image exists on the hardware. Merge `source.psr` first followed by:

   - `source_usb.psr` for USB mode

8. The application will now be running on the hardware.

2.1 To pair and connect the audio source and an audio sink device

1. Turn on the sink device and put it into pairing mode.
2. Plugin\turn on the Audio Source device. Initially the Audio Source device discovers a suitable device, based on RSSI levels, to pair and connect with. The device with the strongest RSSI signal is then connected to.
3. Press and hold the **MFB** button on the audio source device for two seconds to discover and pair further headset devices.
4. Short press the **MFB** button to reconnect to a previously paired device.

2.1.1 To clear the source application's paired device list

In some cases, it is necessary to clear the PDL for successful connection or reconnection. To do this, press the **MFB** button on the Audio Source device for seven seconds.

NOTE Clearing of the PDL only occurs when the Audio Source application is disconnected from all devices. If the Audio Source application is connected to any device, it disconnects before clearing the PDL.

2.2 To set up USB mode on the Audio Source application

This section describes how to set up your PC to use the Audio Source device under USB mode to make a Skype call and play music.

The Audio Source application runs in USB mode by default. Switching between USB and Analog or SPDIF mode is detailed in To set the xIDE project environment.

2.2.1 To install the audio source

1. Insert the Audio Source device into a USB port on your computer.
2. Windows recognizes the hardware and displays the **Found New Hardware** window.
3. Windows enumerates the Audio Source device as a USB composite device (HID and Audio).
4. If prompted by windows for a device driver, select **Install the software automatically**. Restart your PC if necessary.
5. Windows automatically installs the Audio Source device as your Windows Preferred (default) Audio Device.

   Restoring a preferred sound playback and sound recording device
   If you do not want the Audio Source device to be the default sound device for Windows' audio, follow the instructions for the operating system on your PC.
   Windows Vista\7:
6. Open the **Sounds Properties** window.
7. **Start\Control Panel** and click on **Sounds**.
8. Click on the **Playback** tab and select your normal preferred sound playback device.
9. Click **Set Default**.
10. Click on the **Recording** tab and select your normal preferred sound recording device.
11. Click **Set Default**.
12. Click **OK**.

2.2.2 To configure Skype to use the audio source device

1. Open the Skype application.
2. In the **Skype Tools** menu click **Options** and select **General\Audio Settings**:
   a. Microphone source to **USB Audio Device**
   b. Speakers to USB Audio Device
   c. Ringing to USB Audio Device
3. In the Skype Tools menu, click **Options** and select **Advanced\HotKeys**:
   a. Check that **Answer Call** and **Reject/Hang up call** hotkeys are enabled.
   b. Optionally the Ignore call feature can be enabled by setting this to be the Hotkey **ALT+End**.

2.2.3 To make a Skype call

1. Open the Skype application.
2. When a call is initiated using Skype, the Skype ringtone is heard in the headphones.
3. For outgoing calls, the VoIP audio mode must be selected by pressing the **MFB** button on the stereo headset. See for more details on manually switching audio modes.
4. Accept an incoming call using the headset **MFB** button. This closes any open A2DP media channel and opens a SCO audio connection with the Audio Source device.
5. A call can be ended by a long press of the headset **MFB** button, or by pressing a media button on the Headset.

   NOTE For a mono headset, the ringtone plays over SCO and the SCO connection remains open during the call.

2.2.4 To manually switch audio modes

- When the Audio Source device is connected with a stereo headset, manually switch between VOIP SCO audio and Music A2DP audio. This is different behavior from previous Audio Adaptor ADKs.
- When listening to Music on the PC, A2DP can be selected by using the media buttons on the headset. If the VoIP audio is active, pressing a media button on the stereo headset switches to Music mode and ends any active call.
- When using VoIP applications, SCO can be selected by using the **MFB** buttons on the headset. A short press of the headset button has the combined effect of switching to SCO audio and answering any incoming call.
- If VOIP mode is not active when taking VoIP calls, the headset microphone can be disabled.

2.2.5 To configure Windows Media Player to use the audio source device

1. Open Windows Media Player.
2. Go to **Tools\Options…** and select the **Devices** tab.

   NOTE To access the **Tools** menu right-click in the header bar.

3. Select **Speakers** and choose **USB Audio Device** as the **Sound Playback** device.

   NOTE For Windows, select **CSR Audio Source**.

2.3 To set up Analog mode in the Audio Source application

This section describes how to use the Audio Source device in Analog mode to stream audio. In Analog mode, there is no need to set up the PC or portable media players to play music.

The Audio Source application runs in USB mode by default. Switching between USB and Analog mode is detailed in Supporting different input source modes.

2.3.1 To connect the audio source to the music player device

Plug in the input line of Audio Source device to the audio outlet of music player device. For example, for a PC, plug in the input line to the analog speaker output.

2.4 To set up SPDIF mode in the Audio Source application

This section describes how to use the Audio Source device in SPDIF mode to stream audio.

NOTE There is no need to set up the PC or portable music players to play music in SPDIF.

To set up SPDIF mode in the Audio Source application describes how to switch the device into SPDIF input mode.

2.4.1 To connect audio source to the SPDIF device

A Qualcomm® H13191 connector board is required to support SPDIF when using the CSR8670 and CSR8675 development boards. For example, for a PC with a USB soundcard with SPDIF, connect the soundcard SPDIF output to the input on the H13191 connector board.

2.5 To set up I^{2}S mode in the Audio Source application

This section describes how to use the Audio Source device in I^{2}S mode to stream audio.

2.5.1 To connect the Source application to I^{2}S input

The Source application can be configured to accept I^{2}S input using the Configuration Tool. I^{2}S mode can be tested by directly providing I^{2}S input to the development board running the Source Application from a suitable I^{2}S source device. This audio is then streamed over Bluetooth to the connected Sink device.

3 To set the xIDE project environment

Different application builds can be made by changing compile time options. They can then be selected from the Source application **Project Properties** and enabled or disabled as required. These options are described in more detail in:

- Supporting different input source modes
- Supporting DualStream in the Audio Source application
- Enabling button operation in the Audio Source application
- Enabling LED operation in the Audio Source application
- Enabling debug output in the Audio Source application
- Development boards for use with the Audio Source application
- Adding Microsoft Lync support in the Audio Source application
- Including power readings for analog input mode in the Audio Source application
- Configuring A2DP Codecs in the application in the Audio Sourceapplication

NOTE Including project options that are unsupported by the hardware being used may produce the error:

- **Application uses trapsets which firmware doesn't support.**

Execution mode should be left on the setting **Hardware Default** or set to **Assisted Native.** This is the only mode supported on CSR8670 and CSR8675.

3.1 Supporting different input source modes

The Audio Source application supports USB, Analog, and SPDIF input modes. There is different functionality available in USB and Analog/SPDIF modes; as such there is a build control to compile in USB or non-USB functionality.

1. Run xIDE, and open the **source.xiw** workspace.
   <installed directory>\apps\source
2. Amend the **Project Properties** with the required input source. Setting the **Wired Input** option to **Enabled** configures the application for Analog or SPDIF Input mode rather than USB.
3. Load `source.psr` and `source_usb.psr` for USB operation and I2S operations.

   NOTE For wired audio, enable '**wired input'** in the **Project Properties**.

The input mode is configured using the config tool (**Configuration Set>Audio>Input Type**) where the user has an option to confgure different available sources

The input mode can also be determined at run time by changing the User PS Key for USB configuration. If no USB audio is configured, then it falls back to Analog/SPDIF Input mode. However, in this case, the rest of the application may not be configured correctly for Analog or SPDIF mode. For example, the sample rates required to supported Analog, SPDIF, and USB modes may be different.

3.2 Supporting DualStream in the Audio Source application

To support DualStream functionality set the **DualStream** option in the **Project Properties**.

DualStream must then be enabled in the Source application using the ADK configuration tool. To do this, enabled the **Enable Dual Stream** and '**Connect both Devices** options on the **Configuration Set>Bluetooth>DualStream** page of the Source Configuration tool.

The user can specify the Bluetooth address of the 2nd device that the source intends to connect to.

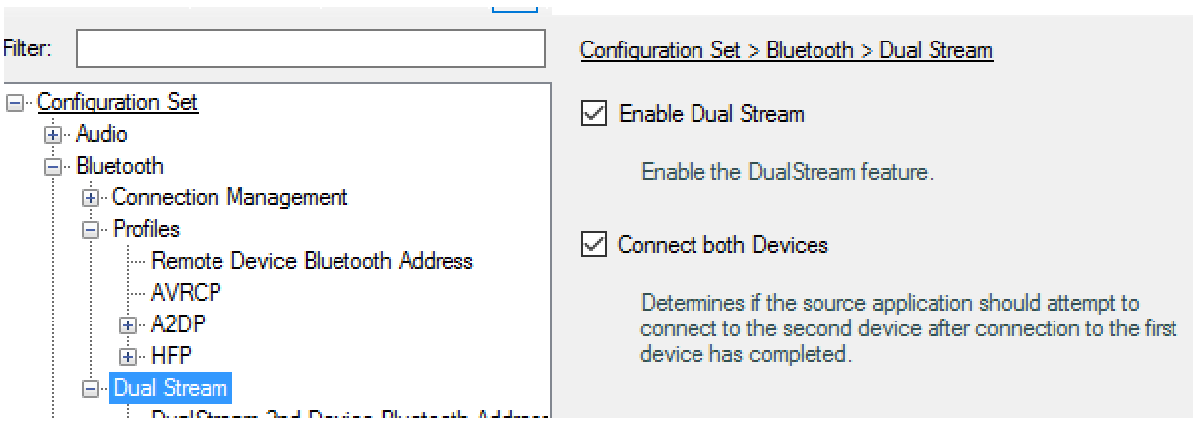

Figure 3-1 Enable Dual Stream in the Source Configuration tool

3.3 Enabling button operation in the Audio Source application

Setting the **Button** option in the **Project Properties** enables button handling in the application.

Button events are created using the **ButtonParsePro** generation tool, where the buttons are defined in `source_buttons.button`.

3.4 Enabling LED operation in the Audio Source application

Setting the **Leds** option in the **Project Properties** enables LED handling in the application.

LED events are created using the **LedParse** generation tool, where the LEDs are defined in `source_leds.led`.

3.5 Enabling debug output in the Audio Source application

Setting the **Debug print** option in the **Project Properties** enables debug output. Debug for particular files can be enabled and disabled by editing the `source_debug.h` file.

3.6 Development boards for use with the Audio Source application

The Audio Source application supports two development boards:

- H13179 with H13374 module (CSR8675)
- CNS10001 (CSR8670)

NOTE By default the ADK **Project Properties** are set to support the CSR8670 development board in USB Mode.

To change the supported board, select the required Qualcomm® BlueCore™ **Hardware Type** and **Flash size** in **Project Properties** (or select **Query chip** with a development board attached).

3.7 Adding Microsoft Lync support in the Audio Source application

Setting the **MS LYNC only** option in the **Project Properties** builds an HFP-only Audio Source and the USB rate will be 16 kHz. This means that no sample rate conversion is needed between the USB rate and Wideband Speech Codec rate.

3.7.1 Changing mic channel from 48 kHz to 16 kHz USB rate in the Audio Source application

The default USB rate is 48 kHz for both Speaker (audio received from USB) and Microphone (audio sent to USB). The Microphone rate can be changed to 16 kHz by setting the **Use MIC 16k** option in the **Project Properties**. The define `USB_MIC_16K` should also be used in the relevant DSP applications.

3.8 Including power readings for analog input mode in the Audio Source application

Setting the **MS LYNC only** option in the **Project Properties** receives power readings from the hardware. The readings can then be configured using PSKEY_USR16. This changes how the `power` library is initialized.

The `source_power` application files can be modified to meet the power requirements.

3.9 Configuring A2DP Codecs in the application in the Audio Sourceapplication

By default the `a2dp` library configures the A2DP Codecs based on the registered Stream End Points. If the application wants to configure the A2DP Codecs instead, set the **App Codec Config** option in the **Project Properties**.

The `A2DP_CODEC_CONFIGURE_IND` message is then sent to the application and the application code can handle this message as required. Example code exists as a guide.

4 USB implementation in the ADK Audio Source application

4.1 USB device enumeration under USB mode in the ADK Audio Source application

When plugged into a laptop/PC using a USB interface, the Audio Source device enumerates as a composite device providing three separate interfaces:

- USB Audio Device

A bidirectional isochronous endpoint providing a 48 kHz 16-bit stereo Sink and a 48 kHz 16-bit mono source. This enables the Audio Source device to present itself as a pair of stereo speakers and a mono microphone.

- HID Compliant Consumer Control Device

An interrupt endpoint enabling the Audio Source device to issue media player control commands to the PC. This enables the Audio Source device to pass AVRCP commands received from an audiosink (such as a headset) to the PC to control a music player capable of understanding Microsoft's standard Media Player control commands. This endpoint also enables Vendor defined commands to be sent to and received from the USB host. The application has basic Vendor behavior implemented, so that a USB host can change the state of the Audio Source application and the Audio Source device can report state back to the host. See for more information on the data sent and received.

- USB Human Interface Device

An interrupt endpoint allowing the Audio Source application to issue key code to the PC. This allows the Audio Source device to provide global hotkey key code to control an application such as Skype.

The interfaces that are supported can be changed by using the USB PS Key (PSKEY_USR9). For example, only unidirectional audio can be implemented or the Consumer Control can be disabled.

4.2 USB host communication in the ADK Audio Source application

The default HID consumer control report descriptor has a Vendor defined section so that Vendor report data can be sent between the Host and Audio Source device, in addition to the standard media commands. This Vendor section can be altered to suit the Vendor requirements.

Currently this interface is used to communicate with the `DongleHost PC` application included with the ADK, so that the HFP v1.7 qualification tests can be run using the Bluetooth SIG PTS tool.

The Interface used in the application is, USB Device to Host\Host to Device Report:

- The Audio Source device sends report data to the USB Host with a report ID of 2.
- The Audio Source device receives report data from the USB Host with a report ID of 2.

The structure of the data sent is as follows:

7	6	5	4	3	2	1	0	Octet
Report ID (2)								0
Command								1
Sub-command								2
Data 0								3

……..

……..

Data 14	17

4.2.1 USB host to device commands

The following Host to Device commands\sub-commands are defined in the ADK Audio Source application:

Host connection (0x00)

Subcommands	Data	Description
Host Disconnected (`0x00`)	None	Indicates that the Host has disconnected
Host Connected (`0x01`)	None	Indicates that the Host has connected

Get\set state (0x01)

Subcommands	Data	Description
Get State (`0x00`)	None	Retrieves the current source state
Enter DFU Mode State (`0x01`)	None	Sets the source state to DFU mode
Enter DUT Mode State (`0x02`)	None	Sets the source state to DUT mode
Enter Inquiry State (`0x03`)	None	Sets the source state to Inquiry mode
Enter Inquiry Scan State (`0x04`)	None	Sets the source state to Inquiry Scan mode

Subcommands	Data	Description
Enter Page State (`0x05`)	None	Sets the source state to Page mode
Enter Page Scan State (`0x06`)	None	Sets the source state to Page Scan mode

AG call state (0x02)

Subcommands	Data	Description
No Call (`0x00`)	None	Informs of no call on the AG
Incoming Call (`0x01`)	None	Informs of incoming call on the AG
Outgoing Call (`0x02`)	None	Informs of outgoing call on the AG
Active Call (`0x03`)	None	Informs of active call on the AG
Call Waiting with Active Call (`0x04`)	None	Informs of active call with call waiting on the AG
Call Held with Active Call (`0x05`)	None	Informs of active call with held call on the AG
Call Held (`0x06`)	None	Informs of call held on the AG

Signal strength (0x03)

Subcommands	Data	Description
Signal Strength Value (`0x00`)	Octet 0: Signal Strength	Sends the signal strength of the AG

Battery level (0x04)

Subcommands	Data	Description
Battery Level Value (`0x00`)	Octet 0: Battery Level	Sends the battery level of the AG

AG Audio (0x05)

Subcommands	Data	Description
Get Audio State (`0x00`)	None	Retrieves the current HFP Audio State
Audio Transfer (`0x01`)	None	Transfers audio request

Network (0x06)

Sub–commands	Data	Description
Operator (`0x00`)	Octet 0 -14: Operator Name	Sends the current operator name
Availability (`0x01`)	Octet 0: Availability	Sends the network availability
Roam (`0x02`)	Octet 0: Roam	Sends the network roam

AG error (0x07)

Subcommands	Data	Description
Invalid Memory Location (`0x00`)	None	Indicates that the memory location is invalid
Invalid Last Number Dial (`0x01`)	None	Indicates the last number dialed is invalid

AG OK (0x08)

Subcommands	Data	Description
Valid Memory Location (`0x00`)	None	Indicates that the memory location is Valid
Valid Last Number Dial (`0x01`)	None	Indicates the last number dialed is Valid
Sent All Current Calls (`0x02`)	None	Indicates that all current calls have been sent

Current call (0x09)

Sub - commands	Data	Description
Current Call Details (`0x00`)	Octet 0: idx Octet 1: dir Octet 2: status Octet 3: mode Octet 4: mpty Octet 5: type Octet 6: number size Octet 7 - 14: number	Sends information of the current call

Voice recognition (0x0a)

Subcommands	Data	Description
Voice Recognition Enable (`0x00`)	None	Enables voice recognition
Voice Recognition Disable (`0x01`)	None	Disables voice recognition

HF indicator (0x0b)

Subcommands	Data	Description
Enhanced Safety Indicator (`0x01`)	Octet 0: Indicator Enable/ Disable	Enables or Disables Enhanced Safety Indicator
Battery Level Indicator (`0x02`)	Octet 0: Indicator Enable/ Disable	Enables or Disables Battery Level Indicator

Link mode (0x0c)

Subcommands	Data	Description
BR/EDR link mode	None	Gets BR/EDR Link mode if Secure Connection or Non-Secure Connection mode

4.2.2 USB device to host commands in the ADK Audio Sour5ce application

The following Host to Device command\sub-commands are defined in the ADK Audio Source application:

Device state (0x00)

Subcommands	Data	Description
Inquiry State (`0x00`)	None	Indicates that the source device is in the inquiry state
Inquiry Scan State (`0x01`)	None	Indicates that the source device is in the inquiry scan state
Page State (`0x02`)	None	Indicates that the source device is in the page state
Page Scan State (`0x03`)	None	Indicates that the source device is in the page scan state
Connected State (`0x04`)	None	Indicates that the source device is in the connected state

Call (0x01)

Subcommands	Data	Description
Call Accept (`0x00`)	None	Accepts call
Call Reject (`0x01`)	None	Rejects call
Call Number Supplied (`0x02`)	Octet 0: Number size Octet 1 - 14: Number	Calls the supplied number
Call Memory Location (`0x03`)	Octet 0: Location size Octet 1 - 14: Location	Calls the supplied memory location

Subcommands	Data	Description
Call Last Number (`0x04`)	None	Calls the last number
Get Current Calls (`0x05`)	None	Gets the list of current calls

HFP Audio state (0x02)

Subcommands	Data	Description
No SLC (`0x00`)	None	No connection exists so audio connection not possible
HFP Audio Disconnected (`0x01`)	None	A connection exists but there is no audio connection
HFP Audio Connected (`0x02`)	None	A connection exists and there is an audio connection

Voice recognition (0x03)

Subcommands	Data	Description
Voice Recognition Disable (`0x00`)	None	Disables voice recognition
Voice Recognition Enable (`0x01`)	None	Enables voice recognition

Link mode (0x04)

Subcommands	Data	Description
No SLC Connection (`0x00`)	None	No SLC connection
No Secure Connection (`0x01`)	None	SLC connection exists but it is not secure connection
Secure Connection (`0x02`)	None	SLC connection exists and it is secure connection

5 DualStream in the ADK Audio Source application

The DualStream feature allows the Audio Source device to stream audio to two A2DP sink devices at the same time, most commonly to two A2DP headsets.

5.1 Using DualStream in the ADK audio source device

Two A2DP sink devices can be paired and connected to the Audio Source device using the procedure detailed in To pair and connect the audio source and an audio sink device. When the two A2DP sink devices are connected, the same audio stream sent by the audio source should be heard in both devices.

The Audio Source application can be configured to automatically initiate a connection to only one A2DP device or attempt a connection to two A2DP devices. This behavior can be changed using the **Connect both Devices** option in the ADK Configuration tool (**Configuration Set>Bluetooth>Dual Stream**).

Alternatively, a second connection can be initiated from the Audio Source device without dropping the first connection by a press of the **MFB** button.

NOTE The audio is suspended during the Audio Source device's attempt to connect to a second device.

5.2 Codecs used in DualStream mode

The only A2DP codecs that can be used when streaming A2DP to two devices at the same time are:

- SBC
- FastStream
- Qualcomm® aptX™

NOTE (1) Both streams must be configured to use the same codec. If a second stream is connected that is configured to use a different codec, then the Audio Source application closes both streams and then re-opens them using the SBC codec.

(2) aptX Low Latency is not supported in DualStream mode.

5.3 Behavior of DualStream implementation

When DualStream mode is enabled the Audio Source device always prefers to take the Master role for both links. This maintains the highest-quality audio.

For the SBC codec, if the Audio Source device gets into a scatternet scenario, the bitpool is dynamically lowered to maintain audio streams that do not experience any drop-outs.

If one of the links experiences a link loss, then the bitpool can be lowered further so that the remaining link can continue with uninterrupted streaming, and not be affected by the transmission occurring on the bad link.

The bitpools used in normal operation, scatternet scenario, and bad link scenario, are all sent from the main application to the DSP, and can be modified by the application engineer, though the default settings do not need changing.

5.4 Limitations of DualStream implementation

This section lists some of the limitations associated with DualStream mode:

- During DualStream operation, both streams must share the same configuration, that is no difference in codecs, sampling rates, and so on The Audio Source application ensures this is the case by reconfiguring the streams if necessary.
- To maintain a high standard of A2DP audio, only Bluetooth EDR devices should be used when two Streams are active. The default behavior is that the Audio Source application checks the capabilities of the remote device on A2DP connection.
- If a second device connects and one of the devices does not support Bluetooth EDR, then the second connection is dropped.
- If a low latency codec is used when streaming to two devices, the limitation is that the Audio Source device must be Master of both links otherwise the audio quality suffers noticeably. This is because of the low latency nature of the codec, with limited buffering on the receiver.
  The low latency codec should also be configured for unidirectional audio only, that is, audio should only be sent from the Audio Source device.

6 States used in the ADK Audio Source application

The Audio Source application defines the following application states. Each Bluetooth profile has its own state machine but this is the main state machine used by the application:

- `SOURCE_STATE_INITIALISING`

Immediately after power is applied to the hardware the Audio Source application enters its Initializing state while the application is set up ready for use. During the initializing state, the PS configuration is read and the Bluetooth Profiles are initialized.

- `SOURCE_STATE_POWERED_OFF`

The Audio Source device can be powered off in Analog or SPDIF mode. This state is used when the hardware is powered on but the device is logically powered off.

- `SOURCE_STATE_TEST_MODE`

This state is entered when the Audio Source application goes into DUT (Device Under Test) mode.

- `SOURCE_STATE_IDLE`

The Audio Source application goes IDLE as a transitional state where it must be decided which state should be entered next. For example, when first powered on the device goes IDLE and then decides which state to go to next, for example go to `SOURCE_STATE_CONNECTABLE` if waiting to connect to a device, or to `SOURCE_STATE_INQUIRING` if inquiring for a device to pair with, etc.

- `SOURCE_STATE_CONNECTABLE`

The Audio Source device is connectable in this state, that is, it allows another device to connect with it as the Adaptor is performing no other action.

- `SOURCE_STATE_DISCOVERABLE`

The Audio Source device is discoverable in this state, that is, it allows another device to pair with it as the Adaptor is performing no other action.

- `SOURCE_STATE_CONNECTING`

The Audio Source device is connecting to another device.

- `SOURCE_STATE_INQUIRING`

The Audio Source device is inquiring for a device to pair and then connect with.

- `SOURCE_STATE_CONNECTED`

The Audio Source device is connected to at least one other device.

7 ADK Audio Source application man machine interface

The man machine interface (MMI) is fully configurable, but has the following default behavior. See the application configuration for the development board in use to determine the PIO/VREG_EN assigned to the relevant button.

- Power On/Off (for Analog or SPDIF mode)
  In Analog or SPDIF mode the **Power On/Off** button is used to perform a power cycle of the Audio Source device.
  If Audio Source device has not been powered on, a **Long** button press (2 second) of the **Power On/Off** button causes the Audio Source application to enter the initializing state and to attempt reconnection or to begin searching for a new device.
  If the Audio Source device has been powered on, a **Long** button press (2 second) of the **Power On/Off** button causes the Audio Source application to disconnect the existing connection (if available) and power off the device.

- MFB
  - Short Press
    Causes the Audio Source application to attempt reconnection to a previously paired device by entering its Connecting state.
  - Long Press
    Causes the Audio Source application to disconnect from a currently connected device (if any) and enter the Inquiring state. It starts searching for a new device to connect with. The **Long Press** duration is 2 seconds.
  - Very Long Press
    When the **MFB** button is held down for 7 seconds, the Audio Source application deletes the PDL.

A PS Key configuration in the ADK Audio Source application

The User PS Keys used by the ADK Audio Source application are described below.

When a PS Key is altered, all words detailed in the PS Key must be set.

The PS Keys used by the Audio Source application can be set using the Source Configuration Tool. The Source Configuration tool can be launched from the Windows **Start** menu:

Start\<ADK installation folder>\Tools\ADK Source Configuration Tool

Open the ADK Configuration Too land select the the appropriate HID option that corresponds to the sourcefrom the dropdown:

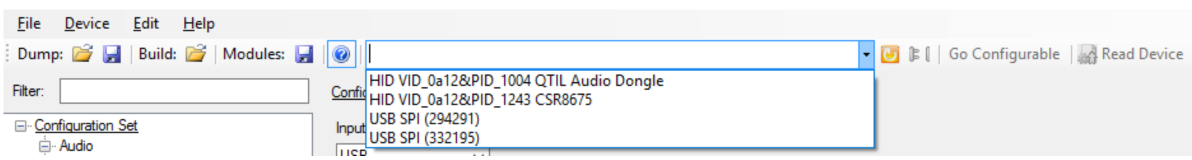

Figure A-1 Select HID option from dropdown

The Source Configuration Tool can read/write source PSR files directly or can be used to read/write the user PS Keys settings to or from the BlueCore device.

The **Read Device** button can be used to read the current configuration from PS, this sets up the Configuration Tool to reflect the current configuration set on the device, from which modifications can then be made. The **Write Device** button on the menu bar can then be used to program the configuration to the development board via the interface selected from the dropdown menu on the menu bar.

NOTE All values stated in this appendix are in Hexadecimal.

PSKEY_USR0: Company ID

States the Company ID.

Word 0

Bits	[15:0]
Purpose	Company ID Upper 16 Bits
Description	Company ID used in the application
Values	Vendor specific

Word 1

Bits	[15:0]
Purpose	Company ID Lower 16 Bits
Description	Company ID used in the application
Values	Vendor specific

PSKEY_USR0 to PSKEY_USR18 are configured in the module `.xml` files and not using the `.psr` files as in previous ADKs. The tables show the details for each of the user configuration items in the Source Configuration tool.

Bluetooth Profiles

Table A-1 Bluetooth Profiles (Audio>Input type)

Size in Bits	2
Purpose	Input Type
Description	Configures the type of audio input to the Audio Source
Values	■ `0` = USB ■ `1` = Analog ■ `2` = SPDIF ■ `3` = I^2S

Table A-2 Bluetooth Profiles (Bluetooth>Profiles)

Size in Bits	1
Purpose	A2DP Bluetooth Profile Support
Description	Enables the A2DP profile
Values	■ `0000` = Disabled ■ `0001` = A2DP v1.2

Size in Bits	1
Purpose	AVRCP Bluetooth Profile Support
Description	Enables the AVRCP profile
Values	■ `0000` = Disabled ■ `0001` = Enable

Size in Bits	2
Purpose	HFP Bluetooth Profile Support
Description	Enables the HFP profile
Values	■ `0000` = Disabled ■ `0001` = HFP v1.6 ■ `0002` = HFP v1.7 **NOTE** The HF Indicator feature and S4 audio settings are enabled by default by the source application, if HFP v1.7 is enabled.

Table A-3 Dual Stream(Configuration Set>Bluetooth>Dual Stream)

Size in Bits	1	1
Purpose	Dual Stream Enable	Connect_both_Devices
Description	Enables Dual Stream if the application has been built with the `INCLUDE_DUALSTREAM` symbol defined	Determines if the source application should attempt to connect to the second device after connection to the first device has completed
Values	`0` = Disabled `1` = Enable	`0` = Disabled autoconnect `1` = Enable autoconnect2

Dual Stream Second A2DP Device address(Bluetooth-Dual Stream>DualStream 2nd device Bluetooth Address)

If Dual Stream is enabled this option in the tool can specify the second Bluetooth address of the second A2DP device that should be connected with when the Connect Policy in the tool (**Connection Management>Reconnect on panic**) is set to **Connect to last device**.

Size in Bits	32
Purpose	Remote Bluetooth Address LAP address
Description	LAP part of Remote Bluetooth Address
Values	`0000` – `00ff`

Size in Bits	8
Purpose	Remote Bluetooth Address UAP address
Description	UAP part of Remote Bluetooth Address
Values	`0000` – `00ff`

Size in Bits	16
Purpose	Remote Bluetooth AddressNAP address
Description	NAP part of Remote Bluetooth Address
Values	`0000` – `ffff`

Remote device Bluetooth address(Bluetooth>Profiles)

Specifies the remote device to connect to when the Reconnect Policy in the configuration tool (**Configuration Set>Bluetooth>Connection Management>Reconnect on panic**) is set to **Connect to Last Device**.

Size in Bits	32
Purpose	Remote Bluetooth Address LAP address
Description	LAP part of Remote Bluetooth Address
Values	`0000` – `00ff`

Size in Bits	8
Purpose	Remote Bluetooth Address LAP address

Description	LAP part of Remote Bluetooth Address
Values	`0000` – `00ff`

Size in Bits	16
Purpose	Remote Bluetooth AddressNAP address
Description	NAP part of Remote Bluetooth Address
Values	`0000` – `ffff`

Bluetooth (Connection Management>Reconnection)

Size in Bits	8
Purpose	A2DP maximum connection attempts
Description	Number of attempts to try and connect A2DP if the remote device is responding
Values	`0` – `ff`

Bluetooth (Connection Management-Pairing)

Size in Bits	16	1	16	16
Purpose	AVRCP maximum connection attempts	Reconnect Policy	Connection maximum retries	HFP maximum connection attempts
Description	Number of attempts to connect AVRCP if the remote device is responding	Determines if the source application should connect to one specific device, or a device from the paired device list	Number of times to try reconnection to the remote device before giving up	Number of attempts to connect HFP if the remote device is responding
Values	`0` – `ff`	`0` = connect to last device only `1` = connect to device from paired device list	`0000` – `fffe` `ffff` = Infinite retries	`0` – `ff`

Used to change the attempted PIN codes when using legacy pairing.

Size in Bits	16
Purpose	Number of PIN codes
Description	Number of different PIN codes to try
Values	`0000` - `00ff`

Size in Bits	16	16	16	16
Purpose	PIN code 0, Digit 0	PIN code 0, Digit 1	PIN code 0, Digit 2	PIN code 0, Digit 3
Description	Digit 0 of the PIN code	Digit 1 of the PIN code	Digit 2 of the PIN code	Digit 3 of the PIN code
Values	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`

Size in Bits	16	16	16	16
Purpose	PIN code 1, Digit 0	PIN code 1, Digit 1	PIN code 1, Digit 2	PIN code 1, Digit 3
Description	Digit 0 of the PIN code	Digit 1 of the PIN code	Digit 2 of the PIN code	Digit 3 of the PIN code
Values	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`

Size in Bits	16	16	16	16
Purpose	PIN code 2, Digit 0	PIN code 2, Digit 1	PIN code 2, Digit 2	PIN code 2, Digit 3
Description	Digit 0 of the PIN code	Digit 1 of the PIN code	Digit 2 of the PIN code	Digit 3 of the PIN code
Values	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`

Size in Bits	16	16	16	16
Purpose	PIN code 3, Digit 0	PIN code 3, Digit 1	PIN code 3, Digit 2	PIN code 2, Digit 3
Description	Digit 0 of the PIN code	Digit 1 of the PIN code	Digit 2 of the PIN code	Digit 3 of the PIN code
Values	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`	`0000` – `ffff`

Size in Bits	1	1
Purpose	Man in the middle	Secure connection mode
Description	Enables man in the middle security	Determines if the connection is secure or not
Values	`0` = Disable `1` = Enable	`0` = `source_no_secure_connection` `1` = `source_secure_connection_mode`
NOTE	Man in the middle feature bit: Enabling this feature bit requires further customer development of the Source application to support Man in the Middle mode. This should be enabled for Test purpose for testing Secure Connection Only Mode.	

A2DP Codecs (Bluetooth-Profiles>A2DP>Codecs)

Used to change the A2DP Codecs in use. The connection preference is used to specify in which order to try the codecs when initiating a media connection, the lower the number the greater the preference (unique numbers should be used for each codec).

Size in bits	2	1	2	1
Purpose	aptX Low Latency Connection Preference	aptX Low Latency Enable	aptX Connection Preference	aptX Enable
Description	Position of aptX Low Latency connection	Adds support for aptX Low Latency Codec	Position of aptX connection	Adds support for aptX Codec
Values	`0` – `7`	`0` = Disabled `1` = Ensabled	`0` – 7	`0` = Disabled `1` = Ensabled

Size in bits	2	1	2	1
Purpose	FastStream Connection Preference	FastStream Enable	SBC Connection Preference	SBC Enable
Description	Position of FastStream connection	Adds support for FastStream Codec	Position of SBC connection	Adds support for SBC Codec
Values	`0` – `7`	`0` = Disabled `1` = Ensabled	`0` – 7	`0` = Disabled `1` = Ensabled

Size in Bits	3	1
Purpose	AptX HD Connection Preference	AptX HD
Description	Position of AptX HD connection	Adds support for AptX HD Codec
Values	`0` - `7`	`0` = Disable `1` = Enable

SBC Configuration

Size in Bits	1
Purpose	SBC Force maximum Bitpool
Description	Stream at the maximum SBC bitpool that is configured
Values	`0` = Use SBC bitpool that is suggested by A2DP library `1` = Use maximum SBC bitpool that is configured

Size in Bits	1
Purpose	Maximum Bitpool
Description	Specifies the maximum SBC Bitpool supported
Values	`0002` - `00fa`

Size in Bits	8
Purpose	Minimum Bitpool
Description	Specifies the minimum SBC Bitpool supported
Values	`0002` - `00fa`

Size in Bits	3
Purpose	Sampling Frequency/Channel Mode

Description	Specifies the supported SBC sampling frequencies and channel modes
Values	`80` = Sampling Frequency 16 kHz * `40` = Sampling Frequency 32 kHz * `20` = Sampling Frequency 44.1 kHz * `10` = Sampling Frequency 48 kHz * `08` = Channel Mode Mono `04` = Channel Mode Dual Channel `02` = Channel Mode Stereo `01` = Channel Mode Joint Stereo * The encoder may not support these sampling frequencies.

FastStream configuration

Size in Bits	1	12
Purpose	Music/Voice Support	Specifies the supported audio directions
Description	Position of AptX HD connection	Specifies the supported audio configuration
Values	`0 - 7`	`01` = FastStream Music Sampling Frequency 48 kHz `02` = FastStream Music Sampling Frequency 44.1 kHz * `03` = FastStream Voice Sampling Frequency 16 kHz * The encoder may not support this sampling frequency.

aptX configuration

Size in Bits	2
Purpose	Specifies aptX Sampling Frequencies supported
Description	aptX Sampling Frequencies
Values	`00` = Sampling Frequency 16 kHz * `01` = Sampling Frequency 32 kHz * `02` = Sampling Frequency 44.1 kHz * `03` = Sampling Frequency 48 kHz * The encoder may not support these sampling frequencies.

aptX Low Latency Configuration

Size in Bits	1	1
Purpose	Bidirectional Audio	Specifies aptX Low Latency Sampling Frequencies supported

Description	Specifies the supported audio directions	aptX Low Latency Sampling Frequencies
Values	`0` = Unidirectional Audio `1` = Bi-directional Audio	00 = Sampling Frequency 44.1 kHz `01` = Sampling Frequency 48 kHz

aptX HD configuration

Size in Bits	1
Purpose	Specifies aptX HD Sampling Frequencies supported
Description	aptX HD Sampling Frequencies
Values	`00` = Sampling Frequency 44.1 kHz * `01` = Sampling Frequency 48 kHz * The encoder may not support these sampling frequencies.

Timers(System>Timers)

Used to change the various timers used by the dongle.

Size in Bits	16
Purpose	Inquiry State Time (secs)
Description	Time to remain in inquiry state
Values	`0000` – `fffe` `ffff` = Infinite

Size in Bits	16
Purpose	Inquiry Idle Time (secs)
Description	Time to remain connectable/discoverable after inquiry state
Values	`0000` – `fffe` `ffff` = Infinite

Size in Bits	16
Purpose	Connection Idle Time (secs)
Description	Time to remain connectable after a failed connection attempt state
Values	`0000` – `fffe` `ffff` = Infinite

Size in Bits	16
Purpose	Disconnect Idle Time (secs)
Description	Time to remain connectable after disconnection, before restarting the connection sequence
Values	`0000` – `fffe` `ffff` = Infinite

Size in Bits	16
Purpose	HFP Connection Failed Time (millisecs)
Description	Time to wait before retrying to connect HFP, after a failed HFP connection attempt
Values	`0000` – `ffff`

Size in Bits	16
Purpose	A2DP Connection Failed Time (millisecs)
Description	Time to wait before retrying to connect A2DP, after a failed A2DP connection attempt
Values	`0000` – `ffff`

Size in Bits	16
Purpose	AVRCP Connection Failed Time (millisecs)
Description	Time to wait before retrying to connect AVRCP, after a failed AVRCP connection attempt
Values	`0000` – `ffff`

Size in Bits	16
Purpose	AVRCP Connection Delay Time (millisecs)
Description	Time to wait before connecting AVRCP, after a successful A2DP connection
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Profile connection delay timer (millisecs)
Description	Time to wait before connecting the next profile when the remote device has initiated the connection
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Link Loss reconnect delay timer (secs)
Description	Time to wait before reconnecting after a link loss. This can be set to be less than the Disconnect Idle Time if a quicker reconnection is required.
Values	`0000` – `ffff` `ffff` = `Infinite`

Size in Bits	16
Purpose	Media Repeat Timer (millisecs)
Description	Time to wait for a repeat of the AVRCP media button pressed event, before treating the media button as released (i.e. for fast forward and rewind events)
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Audio delay timer (millisecs)

Description	Time to wait before connecting audio (which is applied in several cases)
Values	`0000` – `ffff`

Size in Bits	16
Purpose	USB audio active timer (secs)
Description	Time to keep audio active after USB interfaces report that no audio is present
Values	`0000` – `ffff` `ffff` = Always Active Audio

Size in Bits	16
Purpose	Power on connect idle timer (secs)
Description	Time to wait after power on, before initiating the first connection. The value 0xffff can be used to never initiate a connection.
Values	`0000` – `ffff` `ffff` = Never initiate connection from power on

Size in Bits	16
Purpose	Power on discover idle timer (secs)
Description	Time to wait on power on, before initiating the first discovery
Values	`0000` – `ffff` `ffff` = Never initiate discovery from power on

Size in Bits	16
Purpose	BR/EDR `authenticated_payload_timeout_s(N)`
Description	The Authenticated_Payload_Timeout configuration parameter allows the Host to configure the maximum interval between packets containing a MIC received from the remote device when AES-CCM encryption is enabled. Time = N * 10 msec
Values	`0001` – `ffff`

HFP Audio Configuration(Bluetooth>Profiles)

Size in Bits	10
Purpose	HFP Audio Packet Types

Description	The HFP Audio packet types to enable and disable. The non-EDR packet types (HV1, HV2, HV3, EV3, EV4, EV5) are enabled when the values are set. The EDR packet types (2EV3, 3EV3, 2EV5, 3EV5) are enabled when the values are not set.
Values	0001 = HV1 enable 0002 = HV2 enable 0004 = HV3 enable 0008 = EV3 enable 0010 = EV4 enable 0020 = EV5 enable 0040 = 2EV3 disable 0080 = 3EV3 disable 0100 = 2EV5 disable 0200 = 3EV5 disable

Size in Bits	16
Purpose	Bandwidth Upper 16 Bits
Description	The HFP Audio Bandwidth
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Bandwidth Lower 16 Bits
Description	The HFP Audio Bandwidth
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Maximum Latency
Description	The HFP Audio Maximum Latency
Values	`0000` – `ffff`

Size in Bits	16
Purpose	Voice Setting (Air coding)
Description	The HFP Audio Voice Setting
Values	`0000` = CVSD `0001` = u-Law `0002` = a-Law `0003` = Transparent

Size in Bits	2
Purpose	Retransmission Effort

Description	The HFP Audio Retransmission Effort
Values	`0000` = Retransmission disabled `0001` = At least one retransmission, optimize for power consumption `0002` = At least one retransmission, optimize for link quality `00ff` = Don't care

Size in Bits	2
Purpose	Retransmission Effort
Description	The HFP Audio Retransmission Effort
Values	`0000` = Retransmission disabled `0001` = At least one retransmission, optimize for power consumption `0002` = At least one retransmission, optimize for link quality `00ff` = Don't care

USB (System>USB Class of Device)

Used to change the USB configuration of the dongle.

Size in Bits	1	1	1	1
Purpose	USB HID Keyboard Enable	USB HID Consumer Enable	USB Microphone Enable	USB Speaker Enable
Description	Support USB HID Keyboard	Support USB HID Consumer	Support USB Microphone	Support USB Speaker
Values	`0` = Disable USB HID Keyboard `1` = Enable USB HID Keyboard	`0` = Disable USB HID Consumer `1` = Enable USB HID Consumer	`0` = Disable USB Microphone `1` = Enable USB Microphone	`0` = Disable USB Speaker `1` = Enable USB Speaker

B Configuration of LED and button patterns in the ADK Audio Source application

Configuration of LED patterns

An `ledparse.exe` file is included in the ADK. It autogenerates a simple LED control module for the Audio Source application during the project build process. The `ledparse.exe` file included with the ADK installation is found in:

- **<installed directory>\tools\bin**

The LED patterns for the Audio Source ADK, are defined in `source_leds.led` which the **ledparse** tool reads in and generates the source files `source_leds.c` and `source_leds.h`. These source files contain the LED code.

An LED pattern is activated by calling `ledsPlay`. The Audio Source application contains the `source_led_handler.c` file for beginning LED patterns based on state and events:

The `leds_show_state()` function calls `ledsPlay` with the correct pattern for the current state.

The `leds_show_event()` function calls `ledsPlay` with the correct pattern for the generated event.

Configuration of button patterns

A `buttonparsepro.exe` file is included in the ADK. It autogenerates a simple Button control module for the Audio Source application during the project build process. The `buttonparsepro.exe` file included with the ADK installation is found in:

- **<installed directory>\tools\bin**

The Button patterns for the Audio Source application, are defined in `source_buttons.button` which the **buttonparsepro** tool reads in and generates the source files `source_buttons.c` and `source_buttons.h`. These source files contain the Button code.

The Button events are sent to the `button_msg_handler()` function in `source_button_handler.c`, which then act on the button events.

C Implementation details for the ADK Audio Source application

Connection to a remote device

The Audio Source application can be configured in a number of ways to affect how it connects to remote devices.

If the Audio Source device is shipped pre-paired with a headset, then the Bluetooth address of the remote device should be placed in PSKEY_USR7 and PSKEY_USR3 should have the **Reconnect Policy** feature set to `0` (Connect Last Device). The Audio Source device will then always try to connect to this device until a new device is paired and connected.

If the Reconnect Policy feature is set to `0` (Connect Last Device) in PSKEY_USR3 but no address is entered in PSKEY_USR7, then the Audio Source application performs inquiry to search for a device to connect to.

If the Reconnect Policy feature is set to `1` (Connect Paired Device) in PSKEY_USR3, then PSKEY_USR7 is ignored and the Audio Source application searches through the paired device list, trying to connect to each device in turn.

If DualStream is enabled, then PSKEY_USR13 can be used to hold the address of the additional remote device that supports A2DP. This PS Key is used when connecting a second device if the **Reconnect Policy** feature is set to `0` (Connect Last Device) in PSKEY_USR3.

If DualStream is enabled and the Reconnect Policy feature is set to `1` (Connect Paired Device) in PSKEY_USR3, then PSKEY_USR13 is ignored and the Audio Source application searches through the paired device list for the second device to connect with.

The connection behavior can be changed in a number of ways by altering the Timer PS Key (PSKEY_USR8) and Features PS Key (PSKEY_USR3). These configurable features can change for example how long to wait between connection attempts and how many times to retry a connection.

The Audio Source application always connects each profile in the following order: AGHFP, A2DP, AVRCP. A service search is performed on the remote device for each profile before trying to connect, so each profile is attempted every time, if the remote device is in range and responding.

Limitations of analog mode operation

Using the Source application in Analog or SPDIF mode has the following restrictions, which do not apply to USB mode.

- The HFP profile is not intended for use in an Analog or SPDIF mode device. The audio associated with the HFP profile is not routed, so is not audible at the Source application end or the remote end.
- Using a bidirectional A2DP codec results in the audio being heard in one direction only. The audio being sent from the Source application is heard, but the audio arriving at the Source is not routed.

D USB dongle ADK host application - overview

The USB Dongle ADK Host Application is a PC application used to communicate over USB with the source application running on BlueCore hardware. The PC application is used to change the state of the source application and to show the current state of the source application.

To pass HFP v1.7 qualification tests, the source application has to be run against the Bluetooth SIG PTS qualification software. The PC application can also perform test AG functionality to pass these qualification tests.

The Host application searches for a USB device with USB Vendor ID (VID) of 0x0a12, and a USB Product ID (PID) of `0x1004`. A USB device with different VID\PID is specified by writing to registry keys under HKEY_LOCAL_MACHINE\SOFTWARE\CSR\DONGLE ADK 2.0 HOST. Make a new

DWORD value entry with a name of VID with the required USB VID, and another DWORD value entry with a name of PID with the required USB PID.

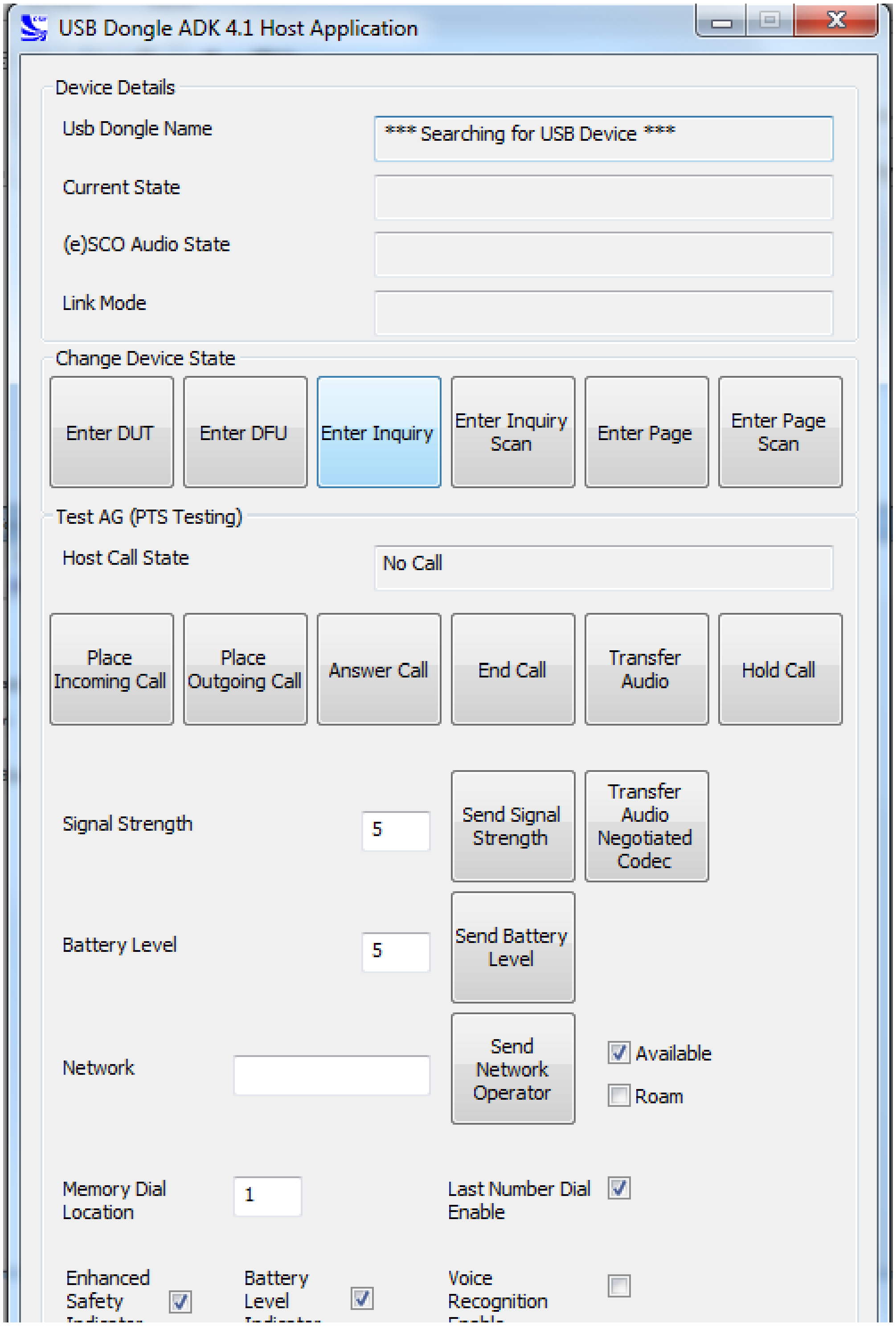

Figure D-1 USB Dongle Host

For a Dongle Host to be run on a 64-bit machine, the key is HKEY_LOCAL_MACHINE\SOFTWARE \Wow64 32Node\CSR\DONGLE ADK 2.0 HOST.

Dongle Host Application has an option for Enhanced Safety Indicator and Battery Level Indicator support. Both the Indicators are enabled by default. The application also has a new tab **Link Mode** added that displays, **Secure Link** or **No Secure Link** after HFP SLC is established on AG side.

NOTE The USB Dongle ADK Host application is included with the ADK in the **tools\bin** subdirectory.

Document references

Document	Reference
USB HID Usage Tables v1.12	www.bluetooth.org

Terms and definitions

Term	Definition
A2DP	Advanced Audio Distribution Profile
ADC	Analog-to-Digital Converter
ADK	Audio or Application Development Kit
AG	Audio Gateway
AIO	Asynchronous Input/Output
AVRCP	Audio/Video Remote Control Profile
BlueCore	Group term for the range of QTIL Bluetooth wireless technology ICs
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
DFU	Device Firmware Upgrade
DSP	Digital Signal Processor
EDR	Enhanced Data Rate
HFP	Handsfree Profile
HID	Human Interface Device
IC	Integrated Circuit
ID	Identifier
LED	Light Emitting Diode
MFB	Multi-Function Button
MIC	Microphone
MMI	Man Machine Interface
PC	Personal Computer
PDL	Paired Device List
PID	Product ID
PIN	Personal Identification Number
PIO	Programmable Input/Output
PS Key	Persistent Store Key
PTS	Protocol Tuning Suite
QTIL	Qualcomm Technologies International, Ltd.
RSSI	Received Signal Strength Indication
SBC	Sub-band Coding

Term	Definition
SCO	Synchronous Connection-Oriented
SIG	(Bluetooth) Special Interest Group
SLC	Service Level Connection
SPDIF	Sony/Philips Digital Interface Format
USB	Universal Serial Bus
VID	Vendor ID
VoIP	Voice over Internet Protocol
xIDE	The QTIL integrated development environment