Qualcomm Technologies International, Ltd.

Mozart 1.0 Entry Level Soundbar Example Design

Application Note

80-CT417-1 Rev. AD

October 25, 2017

Confidential and Proprietary – Qualcomm Technologies International, Ltd.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies International, Ltd. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies International, Ltd.

Qualcomm BlueSuite and Qualcomm Kalimba are products of Qualcomm Technologies International, Ltd. Other Qualcomm products referenced herein are products of Qualcomm Technologies International, Ltd.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. BlueSuite is a trademark of Qualcomm Technologies International, Ltd., registered in the United States and other countries. Kalimba is a trademark of Qualcomm Technologies International, Ltd. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

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
1	AUG 2014	Initial release. Alternative document number CS-00317272-AN.
2	OCT 2014	General Update
3	SEP 2016	Updated to conform to QTI standards; no technical content was changed in this document revision.
AD	OCT 2017	Added to the Document Management System. DRN updated to use Agile number. No change to technical content.

Contents

Tables

Figures

1 QTIL Mozart 1.0 example soundbar

The QTIL Mozart-1.0 Entry Level Soundbar development kit contains everything required for evaluation and development of a 2.1 channel Soundbar with SPDIF and wired analog input, Bluetooth, a wireless Subwoofer and remote control.

The Mozart-1.0 Entry Level Soundbar is based on QTIL’s CSR8670 using ADK 3.5 or later and CSRA65700 ROM devices, both of which support QTIL wireless Subwoofer Audio Transfer technology (SWAT).

The CSR8670 provides the application processor, audio interfaces, Bluetooth functionality and DSP processor in the Mozart Soundbar application.

The CSRA65700 ROM device provides a cost effective audio sink designed for use in wireless Subwoofers.

Information provided on Mozart 1.0 should be read in conjunction with the ADK documentation, appropriate device data sheets, and user guides.

2 Prerequisites for Mozart 1.0

ADK 3.5 or later

The Mozart-1.0 Entry Level Soundbar platform requires an installation of ADK 3.5 or later.

> **NOTE** The ADK installer is available from createpoint and requires a Bluetooth Developers License.

You may be required to complete a Consignee Undertaking before downloading the installer due to the 128-bit encrypted firmware included in the ADK.

For more information visit createpoint or contact your local sales representative.

CSRA65700 downloads

The following downloads are required for use with the CSRA65700 ROM device. The downloads are available from createpoint:

- CSRA65700 Patch Bundle
- CSRA65700 example configuration files for the Mozart-1.0 subwoofer.
- CSRA65700 Subwoofer ROM 1.0 UFE (DSP tuning application)
- Qualcomm® BlueSuite™ PC tools

Documentation

The following documents included with ADKs provide additional useful information.

- *ADK Application Configuration Tool User Guide*
- *ADK Audio Prompts Application Note*
- *ADK Audio Sink Application User Guide*
- *Enabling Qualcomm® aptX™ Codecs in ADK 3*
- *Music Manager Audio Effects E008 User Guide*
- *xIDE User Guide*

The following documents available from createpoint also provide useful information.

- *CSR8670 BGA Data Sheet*
- *CSRA65700 Data Sheet*
- *CSRA65700 User Guide*
- *Mozart-1.0 Entry Level Soundbar Example Design Quick Start Guide*

The following application note included with the CSRA65700 Subwoofer ROM 1.0 UFE is also useful.

- *Wireless Subwoofer Audio Parameters User Guide E200*

3 Getting started with Mozart 1.0

Check that you have all components listed in the *Mozart-1.0 Entry Level Soundbar Example Design Quick Start Guide*. If any components are missing, email sales@csr.com with details of the missing components.

You also need a PC running Microsoft Windows to install the ADK and PC tools on.

3.1 Development hardware for Mozart 1.0

The development boards included in the development kit are shipped with example applications and configurations pre-programmed.

To use the pre-programmed settings to evaluate the Soundbar and Subwoofer, follow the instructions in the *Mozart-1.0 Entry Level Soundbar Example Design Quick Start Guide*.

The development boards have been designed to demonstrate how the PCB for your product could be designed.

Schematics and PCB layout files for the development boards included in the development kit are available from createpoint.

A SPI interface is used to write to, read from and debug CSR8670 and CSRA65700. The CNS10020 debug/programming interface provides the SPI connection between the PC and the development board.

3.1.1 Soundbar example design development board

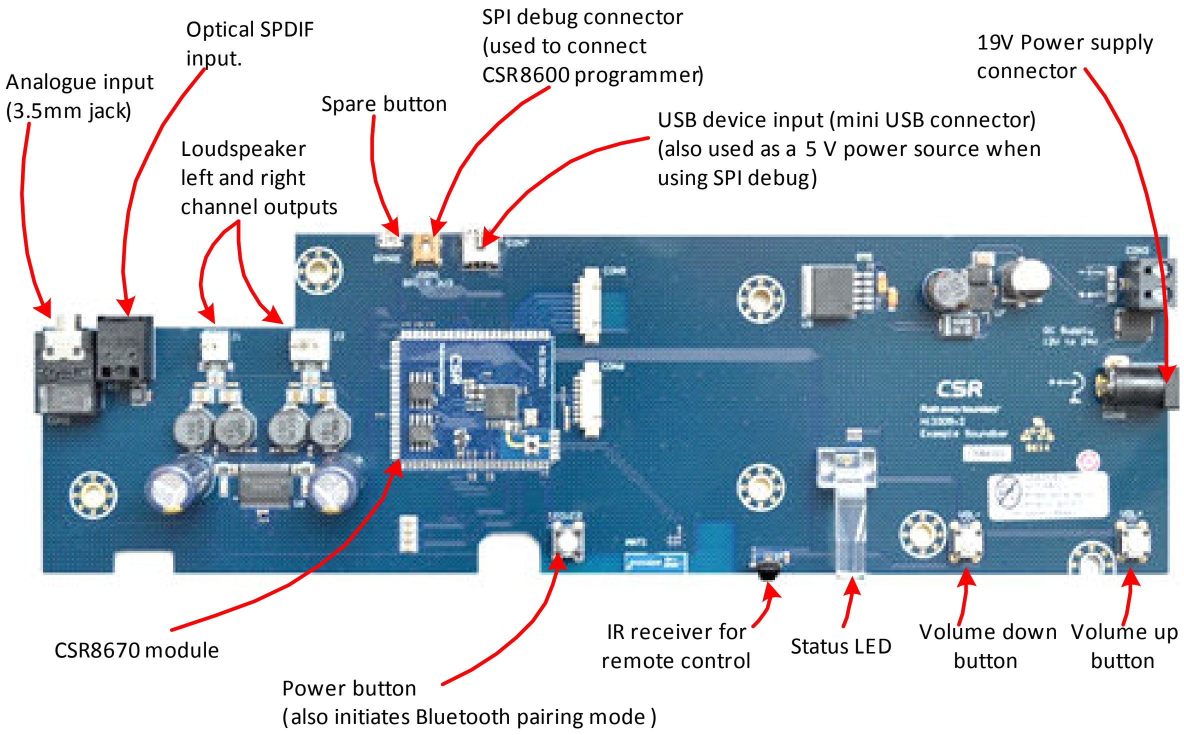

Optical SPDIF input.
SPI debug connector (used to connect CSR8600 programmer)
19V Power supply connector
Analogue input (3.5mm jack)
Spare button
USB device input (mini USB connector) (also used as a 5 V power source when using SPI debug)
Loudspeaker left and right channel outputs
CSR8670 module
Power button (also initiates Bluetooth pairing mode)
IR receiver for remote control
Status LED
Volume down button
Volume up button

Figure 3-1 Mozart-1.0 entry level soundbar example design development board

The Soundbar is made up of the PCBs listed in Table 3-1.

Table 3-1 Mozart-1.0 soundbar PCBs

Board Number	Board Name
H13180	CSR8670 Module
H13305	Soundbar Main Board

Before using the hardware with the PC Tools and the ADK, the board must be connected to the PC. Table 3-1 describes the connections that should be made.

Table 3-2 Soundbar main board (H13305) connectors

Connection	Use
SPI Debug Connector	Must be connected to the CNS10020 debug interface to debug and to read and write flash images and PS keys. The CNS10020 should be connected to a USB port on the PC. It can be disconnected when not required. SPI Debugging requires power to the BlueCore device, either from USB or from the 19 V power supply.
USB device input	Connect to the PC to provide power to the chip and USB functionality.
19 V Power Supply	Supplies power to the main board and module.

Table 3-2 Soundbar main board (H13305) connectors (cont.)

Connection	Use
Loudspeaker Left and Right Outputs	On board amplifier output to connect to your loudspeakers. Pin 1 is positive on connectors J1 and J2. The board is designed to work with two 4-8 Ohm, 25 Wrms loudspeakers.
Optical SPDIF input	Connect to your optical SPDIF source (optional)
Analog input	Connect to your analog audio source (optional)

3.1.2 Subwoofer development boards

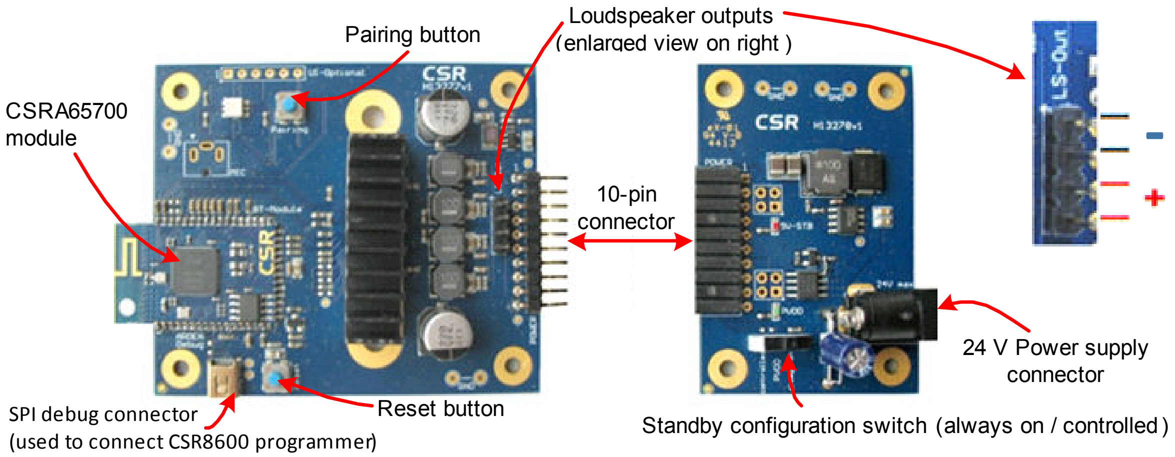


Figure 3-2 CSRA65700 wireless subwoofer development boards

The subwoofer is made up of the PCBs listed in Table 3-3.

Table 3-3 CSRA65700 subwoofer boards

Board Number	Board Name
H13216	CSRA65700 Module
H13277	Subwoofer Main Board
H13278	Subwoofer 24 V to 5 V Power Adaptor Board

Before using the hardware with the PC Tools, the board must be connected to the PC. Table 3-4 describes the connections that should be made.

Table 3-4 CSRA65700 subwoofer connectors

Connection	Use
SPI Debug Connector	Must be connected to the CNS10020 debug interface to read and write PS keys. The CNS10020 should be connected to a USB port on the PC. Can be disconnected when not required. SPI Debugging requires power to the BlueCore device from the Power Adaptor board.
Subwoofer Main Board / Power Adaptor board 10 pin connect	Connect the two boards with the 10 pin connector to provide power to the Subwoofer Main Board.
24 V Power Supply	Supplies power to the 24 V to 5 V Power Adaptor board.
Loudspeaker Outputs	On board amplifier output to connect to your loudspeaker. The board is designed to work with one 4-8 Ohm, 100 Wrms loudspeaker.

3.2 Software for Mozart 1.0

Install the software described on the PC for use with the Soundbar and Subwoofer development boards.

ADK

The Audio Development Kit includes a development environment, tool chain, PC tools, DSP tuning tools, documentation, firmware, and source code for application libraries, DSP libraries, example application, and DSP applications.

Install the ADK and follow the installer prompts to build the libraries. Libraries can also be built after the installation process using the links added to the Windows' **Programs** Menu.

ADK Add-on installers are available for the following extra codecs:

- aptX audio
- AAC
- MP3

These are optional and available from createpoint.

BlueSuite

BlueSuite is a BlueCore development suite designed to help with Bluetooth wireless technology development programs. BlueSuite contains:

- Graphical and command-line applications to configure, test, and program the BlueCore device during design.
- The TrueTest Toolkit, a set of libraries for developing production test applications for BlueCore enabled devices.
- Device drivers to communicate with BlueCore.
- Documentation to help development.

For more details, see the help files included with the installation.

CSRA65700 subwoofer ROM 1.0 UFE

The Universal Parameter Manager application is designed to work as part of a portable testing and tuning system for DSP application. It includes a GUI for setting the parameters of each of the algorithm blocks in the DSP application.

The CSRA65700 Subwoofer ROM 1.0 Parameter Manager is designed to work with the DSP application in the CSRA65700 ROM. The ADK has a separate Parameter Manager designed to work with its DSP applications.

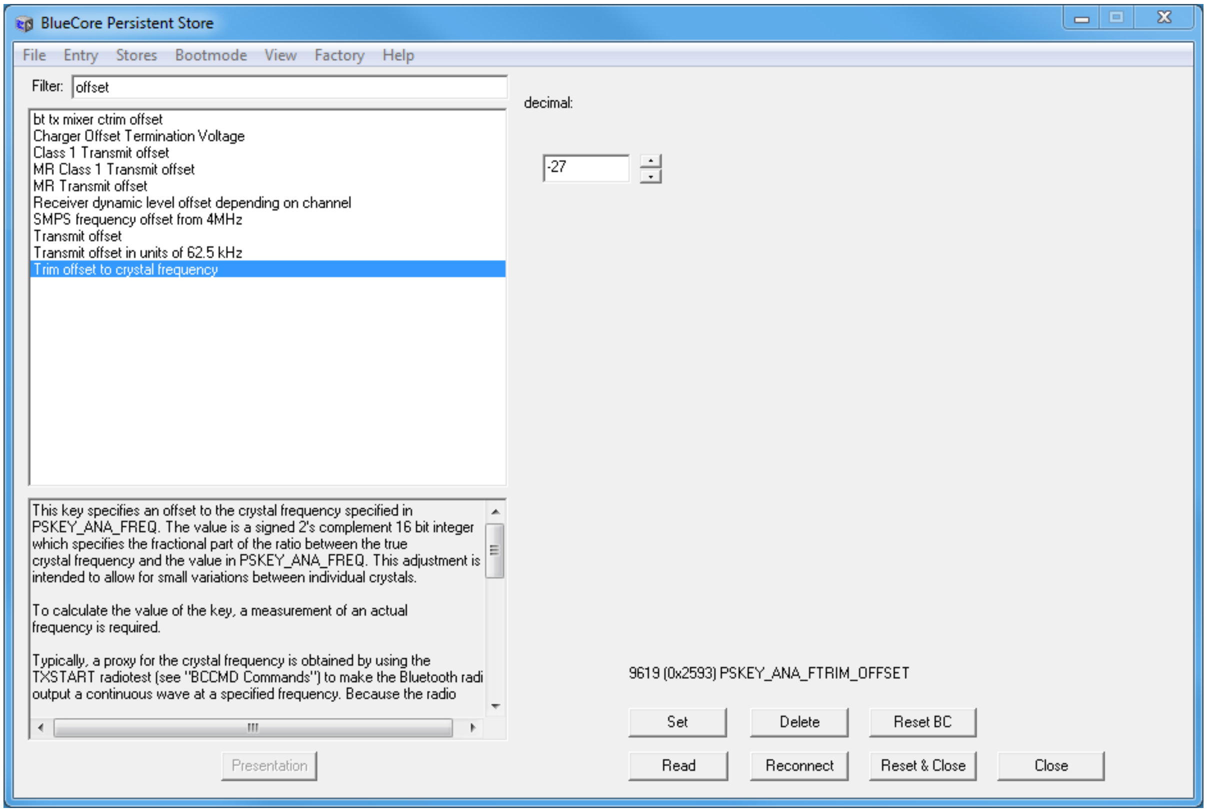
3.3 Mozart development example board specific settings

The development boards included in the development kit have had their crystals calibrated and trim value set as well as Bluetooth Address set. Before programming the board these values should be recorded so they can be restored later.

Correct calibration of the Bluetooth radio is done on a per-device basis on the production line, with the trim value stored in the nonvolatile memory (PS Key). If the trim value is lost, please refer to the device datasheet for information on crystal calibration. The *CSR8670 Data Sheet* and *CSRA65700 Data Sheet* are available from createpoint.

If the Bluetooth address is lost, please use one of the example Bluetooth addresses from CSR documentation or from your own range of Bluetooth addresses.

Back up the Bluetooth address and the Trim offset to crystal frequency PS keys from the Soundbar and Subwoofer with PSTool, as shown in Figure 3-3.



Filter: offset

bt tx mixer ctrim offset
Charger Offset Termination Voltage
Class 1 Transmit offset
MR Class 1 Transmit offset
MR Transmit offset
Receiver dynamic level offset depending on channel
SMPS frequency offset from 4MHz
Transmit offset
Transmit offset in units of 62.5 kHz
Trim offset to crystal frequency

This key specifies an offset to the crystal frequency specified in PSKEY_ANA_FREQ. The value is a signed 2's complement 16 bit integer which specifies the fractional part of the ratio between the true crystal frequency and the value in PSKEY_ANA_FREQ. This adjustment is intended to allow for small variations between individual crystals.

To calculate the value of the key, a measurement of an actual frequency is required.

Typically, a proxy for the crystal frequency is obtained by using the TXSTART radiotest (see "BCCMD Commands") to make the Bluetooth radi output a continuous wave at a specified frequency. Because the radio

decimal: -27

9619 (0x2593) PSKEY_ANA_FTRIM_OFFSET

Figure 3-3 Filtering the PS Key list in PSTool

Table 3-5 lists the keys that should be backed up.

If the default values are displayed, the preset values have either been lost or for some reason PSTool cannot access the external memory device for a ROM device.

Table 3-5 Device specific PS Keys

Friendly Name	Programmer ID	Ordinal	Default Value
Bluetooth address	PSKEY_BDADDR	`0x0001`	`0002 5b 00a5a5`
Trim offset to crystal frequency	PSKEY_ANA_FTRIM_OFFSET	`0x2593`	`0`

3.4 CSR8670 flash image back-up

It is useful to back up the entire flash image store in the CSR8670 internal flash. This includes firmware, application, and PS Key settings and can be used to overwrite any changes to get back to a known state.

To back up the internal flash:

1. Launch the BlueFlash application.
2. Select SPI transport to use.

3. Click **Stop Processor**.
4. Click **Dump**.
5. Save the image dump with an appropriate name.

   The image dump is stored as a `*.xdv *.xpv` pair than can be selected in BlueFlash to Download to a board later.

3.5 CSRA65700 ROM settings back up

All settings for CSRA65700 are stored in PS Keys in an external EEPROM device or SPI Flash device. The development kit uses the EEPROM option.

A complete image dump can be done with the command line tool e2cmd, which is part of BlueSuite. For information type `e2cmd -help` in the command prompt.

A PS Key dump can be done with PSTool.

1. Connection to the device.
2. Select **Store > Implementation** to only include PS Keys stored in the Implementation layer in the EEPROM.
3. Select **File > Dump**.
4. Save the dump psr file with an appropriate name.

   NOTE The process described above does not dump any Bootmode override settings that exist.

   For more information, see the PSTool documentation in BlueSuite.

4 Mozart 1.0 Soundbar application

The Mozart-1.0 Entry Level Soundbar application for CSR8670 is part of the ADK 3.5 Sink application. For full details of the application see the *ADK Audio Sink Application User Guide*.

Before building

If unsure of the application and settings flashed onto the CSR8670, it is advisable to erase the flash before continuing.

> **NOTE** Check the trim value for the crystal and Bluetooth address have been recorded before erasing the flash.

To perform a complete chip erase using BlueFlash:

1. Select SPI transport to use.
2. Click **Stop Processor**.
3. Click **Flash Erase**.
4. In the **Erase Options** window, select **Erase Full Chip** and confirm.

To avoid warnings in the application build process due to failing to write PS Key settings, it is advisable to flash firmware back onto the chip.

This is done using BlueFlash:

1. Select **Choose File** and select the appropriate firmware. The ADK includes CSR8670 firmware in the following location:

   ```
   <ADK> firmware\assisted\unified\gordon
   ```

   The loader and the stack need to be flashed onto the CSR8670, it does not matter which order this is done in.
2. Select either `loader_unsigned.xpv` or `stack_unsigned.xpv`
3. Click **Download**.
4. Repeat steps 2 and 3 for the file not chosen first.

If **Download** is not clickable, or there is an error, check that the development board is correctly connected, that it has power and that the processor has been stopped.

4.1 How to setup application build options for Mozart 1.0

The following steps open and build the application using xIDE:

1. Open the xIDE development environment from the ADK program folder:

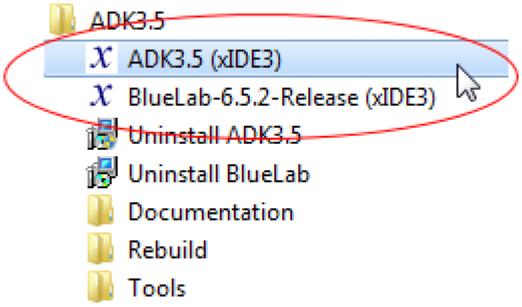


Figure 4-1 Open xIDE from programs menu

2. Select **Open Workspace** from the **Project** menu:

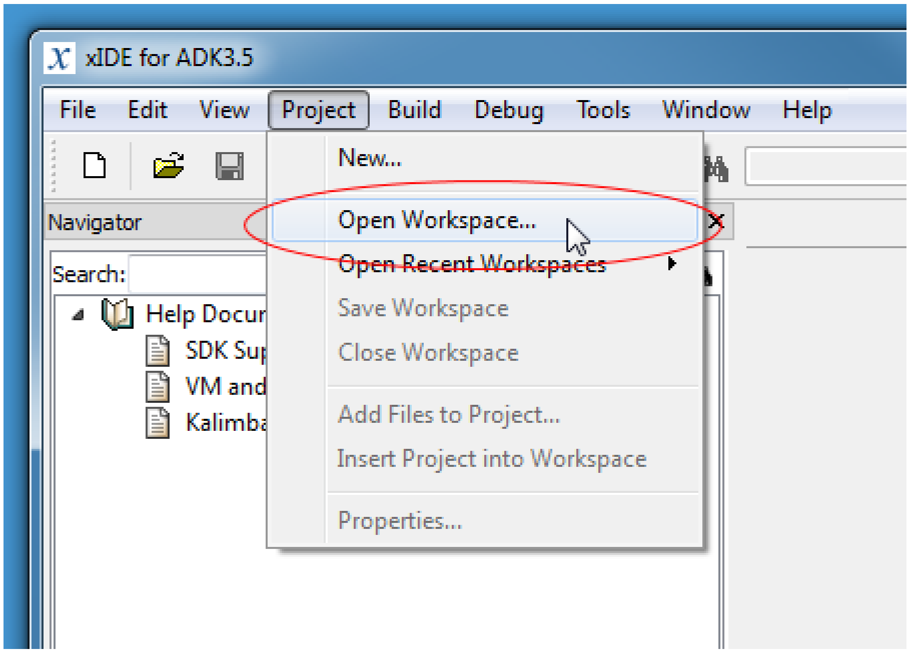


Figure 4-2 Open workspace

3. Select the Sink application workspace from the apps sub-directory

```
<ADK>\apps\sink\sink.xiw
```

4. Select the **Soundbar-Mozart(CSR8670)** project configuration option. This can be done from the main window:

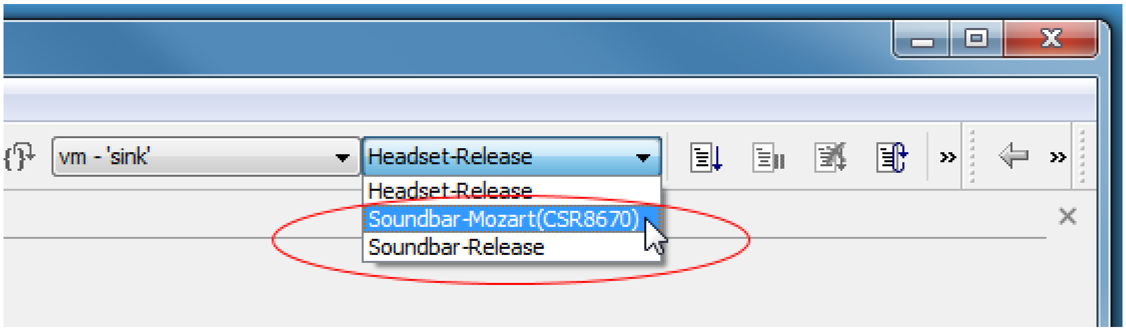


Figure 4-3 Selecting project configuration

It can also be done from **Project Properties**:

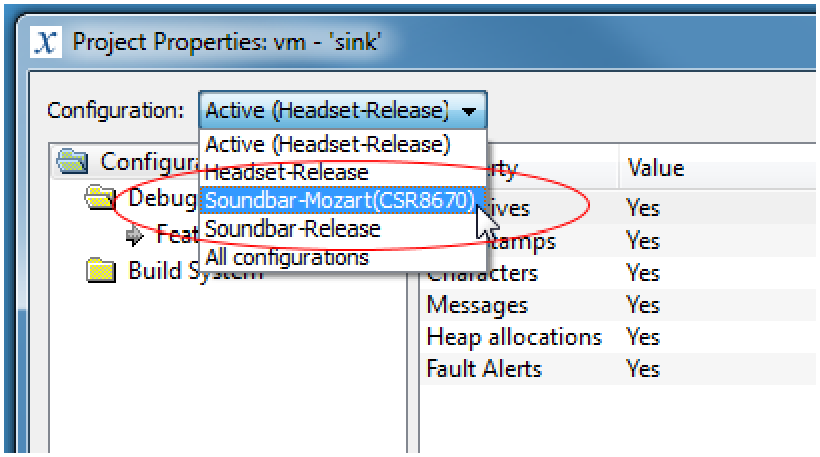


Figure 4-4 Selecting project configuration from Project Properties

The project configuration sets all default build options for the Sink application for the Mozart-1.0 Entry Level Soundbar.

For more information on xIDE, see the *xIDE User Guide*.

For more information on the Sink application, see the *ADK Audio Sink Application User Guide*.

4.2 How to set debug transport for the Mozart Soundbar application

The debug transport must be configured so that it correctly added as part of the build process.

Prerequisites:

Ensure the Debug Transport is set correctly.

- Choose **Debug > Transport** and select the required Debug Transport.

If Debug Transport options are incorrectly set, the application may fail to build correctly.

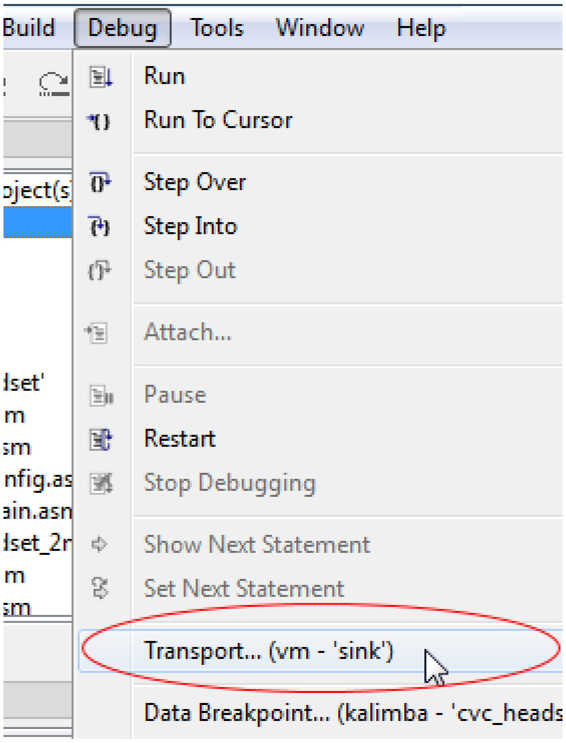


Figure 4-5 Enter debug transport options

4.2.1 Merge application PS Keys for the Mozart 1.0 application

The application requires application configuration setting stored in PS Keys to be present on the CSR8670 for the application to run correctly. These settings include initial run time information and the application configuration.

Use **PSTool** to merge the following PS key files from **<ADK>\apps\sink** onto the device.

- `sink_system_CSR8670.psr`
- `sink_config_Mozart_soundbar.psr`

4.2.2 Build and flash the application to the Mozart development board

To build the active project select **Build Active Project** from the **Build** menu or press **F7** on the PC keyboard.

Alternatively, to build the entire workspace and flash the built image to the development board select **Run** from the **Debug** menu, see Figure 4-6 or press **F5**.

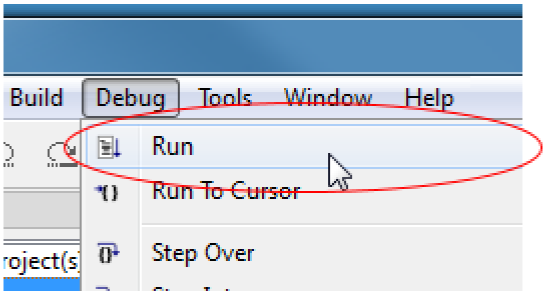


Figure 4-6 Run application

The debugger outputs information in the **Output** window pane.

The application is now running on the hardware assuming there were no errors.

NOTE The application can also run independently of the xIDE environment by selecting **Stop** and resetting the board.

4.3 Mozart 1.0 debug application

xIDE supports the debugging of application code running in the VM of the CSR8670 and the DSP code running on the on-chip DSP.

The application is run on-chip to make the debug environment as similar as possible to the final execution mode of the product.

Common debugging approaches are:

- Setting Break points to enable stepping through code and checking memory/variables.
- Print statements in code are displayed in the Print Channel 0 table of the Output window pane.

Break points can only be set and modified before the application starts or when the application is paused. Memory and variables can only be read when the application is paused. Up to 4 break points can be used for debugging the VM on CSR8670.

Defining the `DEBUG_PRINT_ENABLE` symbol enables debug output. This can be set in the `sink_debug.h` file. Debug for particular files can be enabled and disabled by editing the relevant `#define` in `sink_debug.h`.

Many of the libraries have debug variants that include useful print statements. Refer to the *ADK Audio Sink Application User Guide* for more details.

NOTE When debug is enabled in the application or libraries the application does not run without a debugger attached.

See the *xIDE User Guide* for more information on the use of xIDE.

4.4 Mozart 1.0 application configuration

The application uses configuration settings, stored in PS Keys, to modify its runtime behavior and store parameters used for device operation such as the MMI and feature settings. The application configuration enables the use of the same application in different products with different feature requirements and/or hardware.

ADK sink configuration tool quick start

The ADK Sink Configuration Tool is included with the ADK tools and is linked on the Windows **Programs** menu. Each ADK is tied to a version of Sink Configuration Tool. Errors and unexpected behavior may be seen when the incorrect version of the tool is used.

The following steps allow the application configuration on the development board to be seen. For full details, refer to the *ADK Application Configuration Tool User Guide*.

1. Connect the Soundbar development board to the CNS10020 debug interface and ensure the board has power.
2. Open the **Sink Configuration Tool** located in the Windows **Programs** menu:

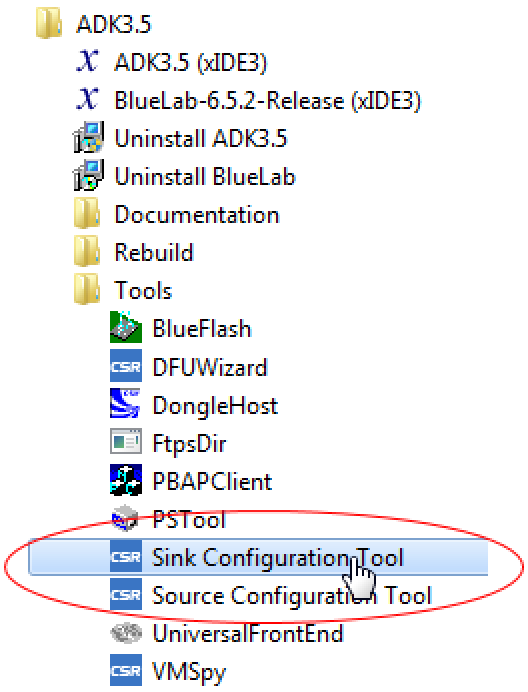
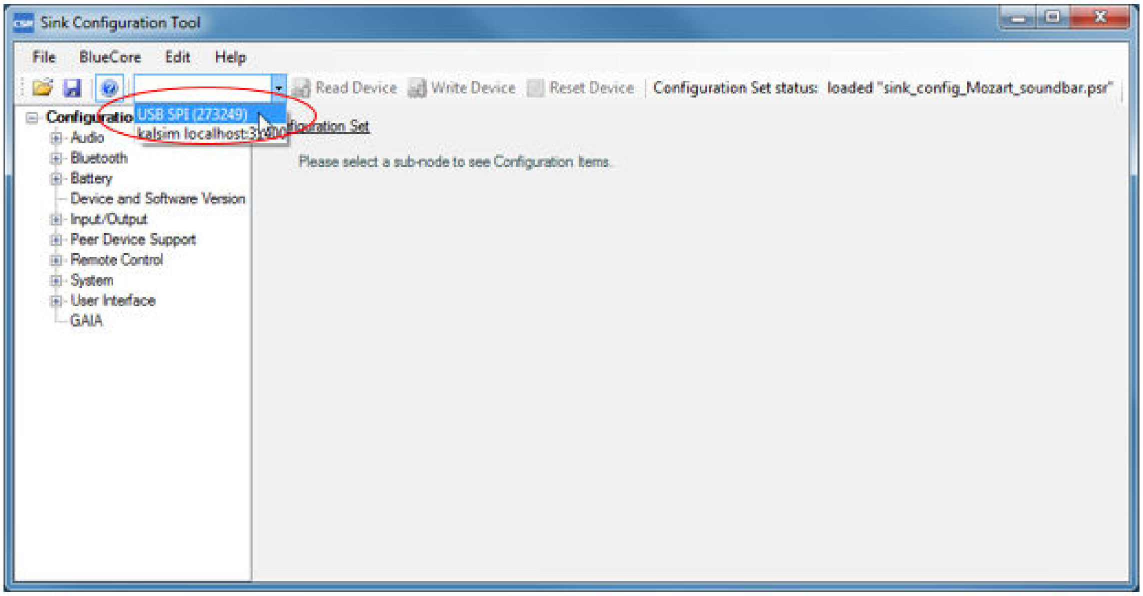


Figure 4-7 Open Sink Configuration Tool from programs menu

3. Load the application configuration used in Merge application PS Keys for the Mozart 1.0 application. This sets up the configuration tool before keys are read from the board and allows for offline configuration changes if required.
   a. In the **Start from an existing Configuration?** option, select **Yes**.
   b. Select the `sink_config_Mozart_soundbar.psr` configuration file and click **Open**.
4. Select the USB SPI device (CNS10020 debug interface) connected to the Soundbar development board:



Configuration Set status: loaded "sink_config_Mozart_soundbar.psr"

Please select a sub-node to see Configuration Items.

Figure 4-8 Select debug interface

On selection of the transport the tool connects to the development board and checks the stored configuration for compatibility. An error is displayed for incompatible configuration versions. The **Read Device**, **Write Device** and **Reset Device** options are now enabled.

5. Click **Read Device** to read the configuration from the development board. If successful, the Configuration Set status changes to **loaded from BlueCore device**.
All application configuration options are accessed through the appropriate node in the Tree View on the left-hand pane. Configuration options are displayed in the right-hand pane with a brief description for each item.
The User Interface node is used to change the MMI configuration settings, see Figure 4-9.

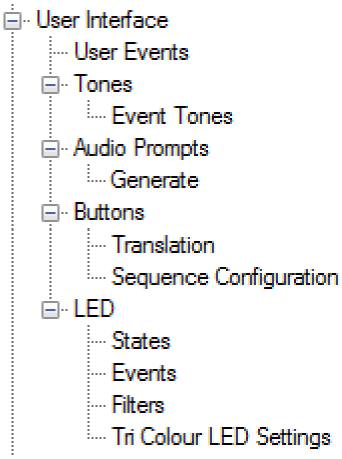

- User Interface
  - User Events
  - Tones
    - Event Tones
  - Audio Prompts
    - Generate
  - Buttons
    - Translation
    - Sequence Configuration
  - LED
    - States
    - Events
    - Filters
    - Tri Colour LED Settings

Figure 4-9 User interface settings tree view

It is possible to add Audio Prompts to replace feedback tones. The default configuration in ADK 3.5 does not include Audio Prompts. To add them, see the *ADK Audio Prompts Application Note*.

The configuration tool cannot make changes to the application in real time. Configuration changes must be written to the CSR8670 and the device must be reset before they take effect.

After making changes to the configuration click **Write Device** to write changes and **Reset Device** to reset the device.

To save the configuration as a PSR file on the PC select **File > Save PSR** or **File > Save PSR As**, as appropriate.

4.5 Mozart 1.0 DSP application

The ADK Sink application is designed to work with several DSP applications that run on the CSR8670 Qualcomm® Kalimba™ DSP. These include music applications and noise/echo cancellation applications.

The ADK Sink workspace includes a typical section of DSP application projects. The Mozart-1.0 Soundbar application is designed to work with the following DSP projects:

- `sbc_decoder`
- `spdif_sink`

The `sbc_decoder` DSP application is used to decoder SBC encoded Bluetooth audio received using the A2DP profile. It is also used to route wired audio. Both the `sbc_decoder` and `spdif_sink` DSP applications output audio to both the stereo speakers and wireless subwoofer and include audio post processing algorithms.

The standard Mozart-1.0 Soundbar configuration does not use the Handsfree Profile which is used for managing call and call audio. The evaluation hardware does not have a microphone so does not need the cVc applications included in the Sink application workspace. The gaming headset applications, `a2dp_low_latency_1/2mic`, are also included in the Sink workspace and are not needed for the Mozart-1.0 Soundbar.

Additional DSP applications can be added for use with other codecs such as aptX, AAC and MP3. Download the appropriate Add-on installer from createpoint, see the *ADK Audio Sink Application User Guide*.

The DSP applications are built along with the VM application in the build process, or can be individually built.

The DSP Applications are written in assembler for the Kalimba processor. The DSP applications can be used without modification and do not require knowledge of assembler to build and run them. For more information, see the Kalimba DSP documentation included with the ADK.

4.6 DSP application tuning

To get the best out of the DSP applications they must be tuned for a specific product.

Tuning options include:

- Parametric EQ
- Compander
- Wireless subwoofer link parameters
- Optional post processing blocks

For more tuning information refer to the *Music Manager Audio Effects E008 User Guide* which includes descriptions of each of the functional blocks as well as how to set the various parameters with the Music Manager section of the Universal Front End parameter manager.

The tools can modify the algorithm parameters of the running system or work in offline mode. To tune the algorithms, CSR recommends that tuning is carried out using the final hardware with SPI connected to the CSR8670. The DSP application needs to be running at the time of tuning, so audio must be playing through the system.

5 Mozart 1.0 wireless subwoofer

The Mozart-1.0 Entry Level Soundbar is designed to use the low cost CSRA65700 ROM device in the wireless subwoofer. The ROM device requires the latest recommended patch bundle from CSR and an application configuration. The recommended patch bundle and example application configuration are available from createpoint.

For general information about the CSRA65700 ROM, see the *CSRA65700 Data Sheet*.

5.1 Mozart 1.0 wireless subwoofer application configuration

The configuration for the application on the CSRA65700 ROM included with the development kit is stored in the device PS Keys in the external EEPROM device. Modification of the keys requires PS Tool. To read and write PS Keys, the board needs to be powered and the PC needs to have SPI access or another appropriate debug transport.

After making any PS Key changes a reset is required to reload the PS Keys into the chip memory.

See the *CSRA65700 User Guide* for the full description and format of the application configuration keys.

5.2 Mozart 1.0 wireless subwoofer DSP application tuning

The DSP application in the CSRA65700 ROM has a number of algorithm blocks that can be modified by changing parameters and setting filter coefficients.

Tuning the subwoofer DSP application follows the same principles as tuning the DSP applications in the soundbar, see DSP application tuning.

For full details on the audio parameters in the DSP application, see the *Wireless Subwoofer Audio Parameters User Guide E200*.

6 Using the Mozart 1.0 soundbar with the subwoofer

The Mozart-1.0 Soundbar connects to the Subwoofer with a low latency Bluetooth link using the proprietary CSR SWAT technology. The Mozart-1.0 Soundbar DSP application includes a configurable filter which separates the low frequencies, from the stereo audio, to send over the SWAT link to the subwoofer.

For information about the Soundbar DSP bass management and associated filtering, see the *Music Manager Audio Effects E008 User Guide*.

The following instructions are for the Soundbar and Subwoofer using default configurations. The behavior may be changed by configuring the devices differently. See Mozart 1.0 wireless subwoofer application configuration and Mozart 1.0 wireless subwoofer DSP application tuning for information about configuration of the devices.

NOTE When the Mozart-1.0 Soundbar is connected to the Subwoofer using SWAT, it can only support one Bluetooth audio (A2DP) connection and cannot support aptX-ll. This is due to Bluetooth bandwidth limitations.

6.1 How to pair the Mozart 1.0 Subwoofer with the soundbar

Before the SWAT link can be established between the Mozart-1.0 Soundbar and subwoofer, the two devices must be paired.

To pair the Subwoofer with the Soundbar:

1. Power on the Subwoofer and Soundbar.
2. Long press the **Pairing** button on the Subwoofer board until the LED toggles quickly between red and blue.
3. Long press (5 seconds) the **Mute** key on the IR remoter control to mute audio and initiate Subwoofer pairing.
4. Short press the **Mute** key on the IR remote control to un-mute the audio.
5. The Subwoofer board LED turns green within about 20 seconds indicating the Subwoofer and Soundbar boards are paired.
6. If the Subwoofer board fails to pair with the Soundbar board, repeat from step 3.
7. The pairing information is stored in the device persistent storage so it is retrievable following a reset.

6.2 How to pair an A2DP source with the soundbar

Before an A2DP link can be established between the Mozart-1.0 Soundbar and A2DP source, the two devices must be paired.

To pair the Soundbar with a phone or other A2DP source:

1. Power off the Soundbar board.
2. Long press the **Power** button on the Soundbar board for 5 seconds to turn it on in Pairing mode.
3. Enable Bluetooth on the phone and search for Bluetooth devices.
4. Find **CSR Mozart Soundbar** on the Bluetooth device list of the phone and connect to it.

7 Controlling the Mozart 1.0 Soundbar

The Soundbar is designed to be controlled by button presses on the Soundbar itself, by IR remote control and by Bluetooth Smart remote control. The default configuration for the Soundbar includes all of these options. Final products may use one or more of these options.

Further details for controlling the Soundbar can be found in the *ADK Audio Sink Application User Guide*.

7.1 PIO/buttons in Mozart 1.0

A typical device uses at least one button to raise or lower a PIO, generating an event in the application.

The default configuration of the Soundbar uses the **Power** button, **Volume Up** and **Volume Down** buttons. A PIO is also used to detect wired analog audio attachment, which can be used to automatically switch the audio routing to use the analog input.

The configuration can be seen and modified in the **User Interface > User Events** node in the Sink Configuration Tool. See Mozart 1.0 application configuration for more information about changing the configuration.

7.2 IR remote control in Mozart 1.0

The Mozart-1.0 Soundbar includes hardware and software to work with the IR remote control included in the development kit.

7.2.1 Enabling the IR remote in Mozart 1.0

To enable the supplied IR Remote Control, it needs to be set to the same IR code set (1561) as used in the Mozart-1.0 Soundbar. To do this:

1. Press and hold the preset key **tv** or **stb** until the LED flashes twice.
2. Then enter the IR code 1561.
3. The LED flashes twice again to confirm the action.

The **tv** or **stb** preset key is now configured with code 1561. If a different preset key e.g. **dvd** is pressed afterwards then the IR transmits a different IR code. This means the IR does not work with the soundbar until the preset key configured for the Soundbar is pressed.

7.2.2 Resetting the IR remote in Mozart 1.0

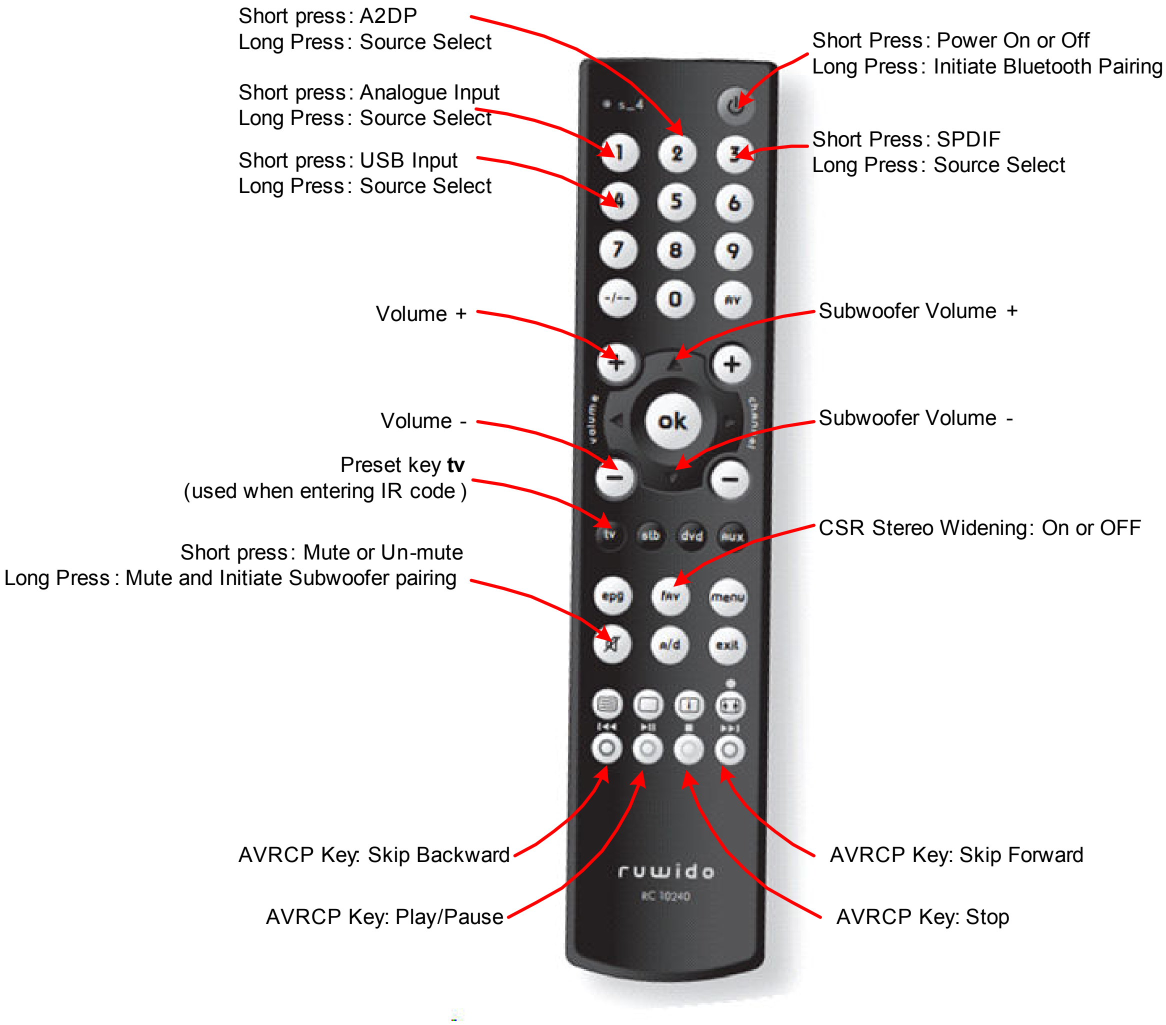
To reset the remote control to the factory default:

1. Long press the preset key configured for the Mozart-1.0 Soundbar until the LED flashes twice.
2. Enter 0000.
3. The LED flashes twice to confirm the setting.

The IR remote control is now reset to factory default.

7.2.3 Default IR remote control button mappings in Mozart 1.0

Figure 7-1 shows the button mappings for the IR remote control supplied with the Mozart Development Kit.



Short press: A2DP
Long Press: Source Select

Short press: Analogue Input
Long Press: Source Select

Short press: USB Input
Long Press: Source Select

Volume +

Volume -

Preset key **tv**
(used when entering IR code)

Short press: Mute or Un-mute
Long Press : Mute and Initiate Subwoofer pairing

AVRCP Key: Skip Backward

AVRCP Key: Play/Pause

Short Press: Power On or Off
Long Press: Initiate Bluetooth Pairing

Short Press: SPDIF
Long Press: Source Select

Subwoofer Volume +

Subwoofer Volume -

CSR Stereo Widening: On or OFF

AVRCP Key: Skip Forward

AVRCP Key: Stop

Figure 7-1 IR remote control button mappings

The configuration can be seen and modified in the **Remote Control** node in the Sink Configuration Tool, see Mozart 1.0 application configuration.

7.3 Bluetooth Low Energy remote control in the Mozart 1.0 Soundbar

The Mozart-1.0 Soundbar can be controlled by a Bluetooth Smart remote control that supports the HID over GATT profile.

The soundbar configuration can be seen and modified in the **Remote Control** node in the Sink Configuration Tool. See Mozart 1.0 application configuration for soundbar configuration details.

NOTE The Mozart-1.0 Soundbar has been designed to work with one Bluetooth Low Energy remote control. Connecting more than one Bluetooth Smart remote control may cause poor audio performance due to Bluetooth bandwidth limitations.

For further information about using this feature, visit createpoint or contact your local sales representative.

8 Audio sources in Mozart 1.0

The Mozart-1.0 Soundbar can be configured to use up to four different audio input methods. The default configuration enables all input methods.

To switch between audio sources, the on-chip application needs to receive an input event. The event can be triggered from a new connection, or can be triggered from a button press or remote control input. See Mozart 1.0 application configuration for details about controlling the soundbar.

The audio source priority in the Soundbar is:

1. Bluetooth Audio
2. USB Audio
3. Wired Audio

Audio source priority can be changed in the application code.

For more information, see the *ADK Audio Sink Application User Guide*.

8.1 Bluetooth audio in the Mozart 1.0 Soundbar application

The Mozart-1.0 Soundbar connects to a Bluetooth audio source devices using the A2DP profile. Depending on which codecs have been built into the Soundbar application, one of the following codecs is used to receive the audio:

- SBC - The mandatory A2DP codec, supported by all A2DP sources.
- MP3 - An alternative A2DP codec.
- AAC - The audio codec used for A2DP by Apple iOS devices (SBC also available with Apple iOS devices).
- aptX - Qualcomm®'s high quality A2DP codec, supported by many Android devices. See http://www.aptx.com for more details.

Whenever possible, the Soundbar always selects the highest quality available audio codec.

The Soundbar controls audio playback using the AVRCP profile. This enables Play, Pause, track skip and volume control by the remote control connected to the Soundbar or a button press on the Soundbar.

8.2 USB audio in the Mozart 1.0 Soundbar application

The Soundbar enumerates as a stereo USB audio device, allowing audio to be played from a PC. The Soundbar also supports volume control using HID consumer transport control.

8.3 Wired analog audio in the Mozart 1.0 Soundbar application

The Soundbar uses the two ADCs on the CSR8670 to enable wired analog audio input for the Soundbar. The Soundbar in the development kit has hardware connecting to a PIO to trigger an event on wired audio connection. This event can enable switching the audio routing to the wired analog audio input on connection.

8.4 SPDIF in the Mozart 1.0 Soundbar application

The CSR8670 supports Optical and Coaxial SPDIF depending on external components used. For example circuits and more information, see the *CSR8670 BGA Data Sheet*.

The Mozart-1.0 Soundbar development board uses Optical SPDIF input and supports all standard SPDIF sample rates. For more information, refer to the *ADK Audio Sink Application User Guide*.

Document references

Document	Reference
ADK Configuration Tool User Guide	80-CT554-1/CS-00401879-UG
ADK 4.3 Source Configuration Tool User Guide	80-CF419-1/CS-00406798-UG
ADK Audio Prompts Application Note	80-CT418-1/CS-00237358-AN
ADK Audio Sink Application User Guide	80-CT439-1/CS-00236868-UG
CSR8670 BGA Data Sheet	80-CE518-1/CS-00127997-DS
Enabling Qualcomm aptX Codecs	80-CT446-1/CS-00314161-AN
Qualcomm BlueCore Music Manager User Guide (E00F)	80-CF438-1/CS-00406876-UG
xIDE User Guide	80-CT405-1/CS-00101500-UG

Terms and definitions

Term	Definition
A2DP	Advanced Audio Distribution Profile
ADC	Analog-to-Digital Converter
ADK	Audio Development Kit
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
CSR	Cambridge Silicon Radio
DSP	Digital Signal Processor
EEPROM	Electrically Erasable Programmable Read Only Memory
EQ	EQualiser
HID	Human Interface Device
iOS	iPhone Operating System
IR	Infrared
LED	Light Emitting Diode
MMI	Man Machine Interface
PC	Personal Computer
PCB	Printed Circuit Board
PIO	Programmable Input/Output
PS	Persistent Store
QTIL	Qualcomm Technologies International, Ltd.
ROM	Read Only Memory
SBC	Sub-band Coding
SPDIF	Sony/Philips Digital Interface Format
SWAT	SubWoofer Audio Transfer
UFE	Universal Front End
USB	Universal Serial Bus
VM	Virtual Machine
xIDE	CSR's Integrated Development Environment