.

CC256x Advanced Voice and Audio Features

This page describes the advanced audio features of the CC256xB Bluetooth device in addition to the normal voice and audio features required by the Bluetooth® 4.0 specification. This document describes new system features such as Advanced Audio Distribution Profile (A2DP) and Wideband Speech by utilizing the hardware flexibility of the PCM and I2S™ interfaces.

Contents

Introduction

There have recently been significant advancements in both new audio features and new requirements for cell phones and smartphones, Bluetooth, and FM shared function devices. These technological steps are easily seen in the new profiles for Wideband (WB) Speech and A2DP stereo processing for Bluetooth.

The CC256xB contains various functional blocks including Bluetooth, BLE, and RF block. However, not all of these functions are used all of the time. In particular, the BLE processor is idle when BLE is not being used, and the processor is instead used to support the new audio requirements. In this mode, it is known as the Audio and Voice Processor (AVPR).

This document describes the advanced audio features of the CC256xB devices, and specifically describes the new AVPR functions.

💡**Note:** This page describes new audio features of the CC256xB dual mode Bluetooth controller, a new ROM spin of the CC256xA device. Apart from device label, you can verify the correct part number by making sure the LMP subversion for CC256xB is 0x1B90(instead of 0x1B0F for CC256xA) by issuing the HCI_Read_Local_Version_Information standard HCI command before loading the Service Pack.

Assisted A2DP (A3DP)

The CC256xB device contains support for A2DP that significantly simplifies the system design and reduces host processing requirements.

A2DP Requirements

- Mono, stereo, or dual channel
- ACL packets (DM3/5, DH3/5)
- L2CAP framing
- SBC encoding/decoding
- Required sampling rates: 44.1 Ksamples/s and 48 Ksamples/s
- Maximum bit rates: 512 Kbits/s stereo, 320 Kbit/s mono

These requirements impose significant processing demands on the host.

Traditional A2DP

The host must implement the audio decoding capability and complete the A2DP path: SBC encoding, L2CAP framing, and a continuous Bluetooth ACL link datastream to the internal Bluetooth controller. This approach adds significant additional overhead to the host.

Bandwidth-Dependent Bit Pool

The supported range of bitpool values for the A2DP stream (the specified limits are 2 to 250) are negotiated as part of the A2DP signaling procedure, which is executed by the host. The bitpool, together with other codec parameters, determines the bitrate of the entire stream.

Table 1 summarizes the recommended sets of SBC parameters in the A2DP specifications.

Table 1. Bandwidth Dependency on Bitpool Values

SBC Encoder Settings[1]	Mid-Quality				High-Quality			
	Mono		Joint Stereo		Mono		Joint Stereo	
Sampling frequency (kHz)	44.1	48	44.1	48	44.1	48	44.1	48
Bitpool value	19	18	35	33	31	29	53	51
Resulting frame length (bytes)	46	44	83	79	70	66	119	115
Resulting bit rate (KBits/s)	127	132	229	237	193	198	328	345

[1] Other settings: Block length = 16, allocation method = loudness, sub-bands = 8.

The bandwidth-dependent bitpool provides audio quality adjustment as result of bandwidth availability. When RF performance degrades, or when bandwidth is limited as result of other traffic or activity, the bitpool decreases adaptively to the minimum negotiated value. It then increases back to the set optimal value when there are no channel constraints. The negotiated bitpool range is configured by the HCI_VS_A3DP_Codec_Configuration and the HCI_VS_A3DP_Multiple_SNK_Configuration commands.

Sample Rate Converter

According to the A2DP specification, the source role shall support at least one of the two sampling frequencies, 44.1 kHz and 48 kHz (see the specification definitions for SBC sampling frequency). However, audio on the host file system may be saved with a sampling frequency of 8 kHz, 11.025 kHz ,12 kHz, 16 kHz, or other rate. Table 2 shows the supported sampling frequencies for A2DP.

Table 2. A2DP Supported Sampling Frequencies

Position	Sampling Frequency (Hz)	Support in SRC[1]	Support in SNK
Octet 0; b7	16,000	O	O
Octet 0; b6	32,000	O	O
Octet 0; b5	44,100	C1	M
Octet 0; b4	48,000	C1	M

[1] C1: At least one of the values shall be supported.

In order to ease the feature integration, the 65-nM sample rate converter converts a given audio sampling frequency to one of the mandatory A2DP frequencies. The host, however, can transfer the audio to the Bluetooth Controller on the PCM bus in its original sampling frequency. Figure 16 shows the sampling rate conversion system overview.

Sampling Frequencies Conversion

The table below details the sampling frequencies conversions. The sampling rate conversion is configured in the HCI_VS_A3DP_Codec_Configuration command.

Sampling Frequency Conversions

Input Frequency (Hz)	Output Frequency (Hz)
8,000	48,000
11,025	44,100
12,000	48,000
16,000	48,000
22,050	44,100
24,000	48,000
36,000	48,000

Enabling and Disabling A3DP

To enable A3DP, send the HCI_VS_AVPR_Enable 0xFD92 command with the appropriate parameters.

Opening a Stream

To open a stream, the protocol stack should send the open stream command with the appropriate parameters.

The codec configure command cannot be sent before the stream is open; therefore, it may be useful to send a codec configure command immediately after sending an open command if the stack does not receive any further indications of the initial configuration.

It is important to note that issuing the open stream command to the controller is asynchronous; that is, the protocol stack must wait for a command complete indication from the controller before the stream can be considered open. The introduction of asynchronous steps into the open stream process may introduce some implementation issues, such as when to notify higher (application) layers and how to handle incoming events if the stream is not yet open. It is up to the protocol stack implementer to handle such issues.

Configuring a stream

· To configure a stream in a single SNK scenario, the host should send the codec configure command. This command defines the SBC parameters in addition to the incoming PCM parameters. The controller will use sample rate conversion if it is required; that is, when the negotiated sample frequency does not match the incoming PCM sample frequency. It is the host responsibility to know the PCM sample frequency and to negotiate a sample frequency for SBC which either does not require sample rate conversion or can be handled by the sample rate converter.

The codec configuration is not sent when a stream is in a streaming state. It must be sent upon initial configuration (when the stream is first opened) and whenever a re-configuration occurs for changing PCM or SBC parameters.

Starting a stream

To start a stream, the host should send the start stream command. Note that after the stream has started, the codec configuration cannot be altered. Only bitpool values may be changed using the multiple SNK configuration command in a multiple SNK scenario. In order to change the stream configuration, the stream must first be suspended (using the stop stream command), reconfigured, and than started again.

Once a stream starts, it is recommended to use an orderly manner for closing the stream—first, suspend the stream and then, close it.

The start stream command introduces asynchronicity into the configuration process because the protocol stack must wait for a command complete indication to consider the stream streaming.

Suspending a stream

To suspend a stream, the host should send the stop stream command.

The stop stream command also introduces asynchronicity into the configuration process; the protocol stack must wait for a command complete indication to consider the stream open.

Closing a stream

To close a stream, the host should send the close stream command.

The close stream command introduces asynchronicity into the configuration process as well, because the protocol stack must wait for a command complete indication to consider the stream closed.

Software Flow Diagram

The following diagram shows the A3DP signaling sequence from the host(MSP430 in this case) to the CC256xB Bluetooth controller.
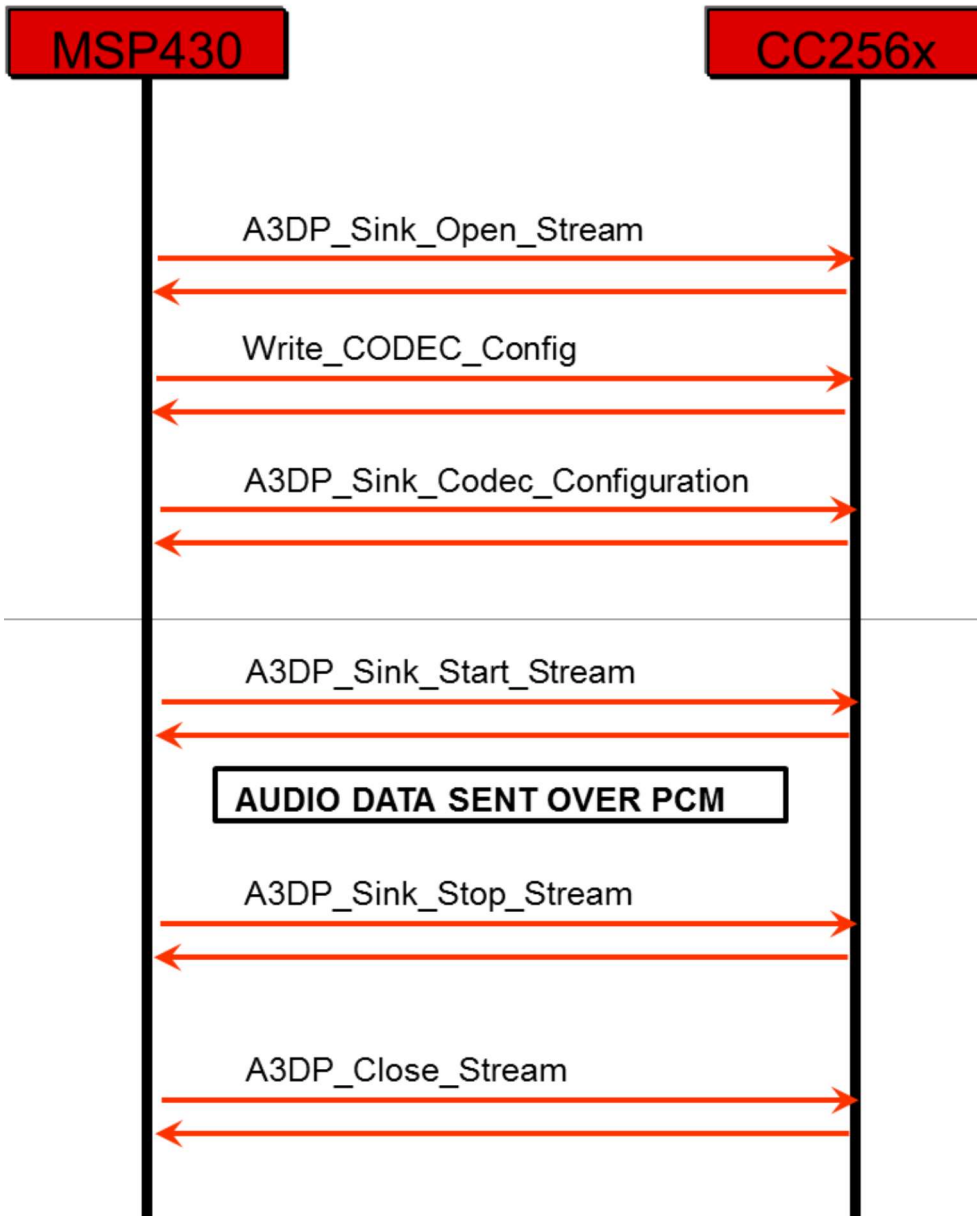
Assisted A2DP

In this approach, the host implements only the audio source decoding capability and PCM signaling as before, then establishes a Bluetooth link and negotiates the L2CAP parameters. Thereafter, the CC256x device performs all required PCM to A2DP processes on an ongoing basis.

[insert pic here]

💡**Note:** This process is transparent to the host; the standard PCM interface is all that is required, whether the headset is directly connected to the phone or a wireless handset via Bluetooth.

Assisted A2DP Features and Description

Assisted A2DP (A3DP) has these features:

- Partitioning of audio stream processing
    - Host: Negotiation (parameters, L2CAP channels, etc.) and remote control (AVRCP)
    - Bluetooth: SBC encoding, as well as AVDTP and L2CAP packaging
- Only one A2DP stream is supported using this feature
- Transparent to audio system: Uses the existing PCM/I2S interface
- CC256xB assisted A2DP saves 15 MIPS to 30 MIPS in the host processor

[insert pic: A3DP System Overview]

Limitations

The following table summarizes known limitations of current A3DP implementation:

Known Limitations of A3DP

Item	Limitation
AVDTP service type	The only supported AVDTP service type is the basic type.
L2CAP Flush	L2CAP channels for the same connection handles of the A3DP connection handles would always be non-flushable.
L2CAP mode	The only supported L2CAP mode of operation is Basic Mode.
A2DP Content Protection	A2DP Content Protection is not supported.
SBC Frame Fragmentation	SBC frame fragmentation is not supported. As a result, the minimum negotiated L2CAP MTU of the peer device is limited to 260.
SBC bitpool	Bitpool maximum supported value is 57.
SBC Sub-bands	Eight only

WB Speech

CC256xB devices contain support for WB speech that significantly simplifies the overall system design and reduces host processing requirements. Specifically, the devices fully support WBS as defined in the Bluetooth Hands Free Profile (HFP 1.6) Specification.

WB Speech Compared to Standard Bluetooth Voice

The mSBC Bluetooth air encoding allows twice the voice quality at the same air bandwidth. mSBC is a slightly modified form of the standard SBC specification that Bluetooth uses.
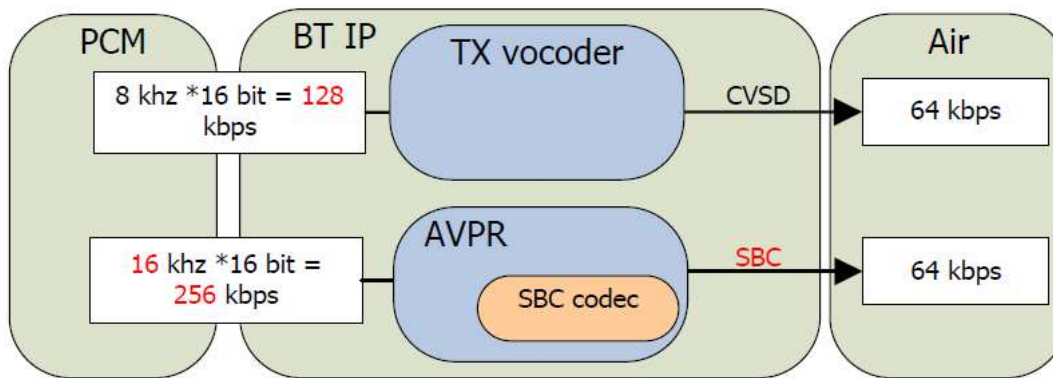
WBS Features in CC256x devices

- Provide mSBC encoding and decoding in the CC256xB device instead of the host(to offload the host processor)
- Support legacy headsets (non-mSBC-compliant, 8 Ksamples per second) as well as WB speech headsets, all managed transparently to the phone system. This capability allows the phone to have a standard interface for all headsets.
- Provide support for packet loss concealment (PLC) for WBS
- The internal mSBC codec can be applied only through the PCM interface. Assisted WBS over HCI (UART) is not supported.

Note: Only one WB speech extended synchronous connection oriented (eSCO) is supported at a time.

Software Setup for Assisted WBS

1. The AVPR functions cannot run concurrently with BLE or ANT. You must first disable BLE/ANT with the corresponding command:
   - BLE: `Send_HCI_VS_BLE_Enable 0xFD5B, 0, 0`
   - ANT: `Send_HCI_VS_ANT_Enable 0xFDD0, 0, 0, 0`
2. The `HCI_VS_Write_CODEC_Config (0xFD06)` command is required to set fSYNC to a 16-kHz sampling rate (and correspondingly double the clock speed), as well as set up the standard codec PCM parameters.
3. The following commands are required to activate WB speech in the CC256xB device:

```
//configure codec to 16 kHz 16 bit samples. will be SBC encoded and sent over 64 kbps. eSCO link.
HCI_VS_Write_CODEC_Config 0xFD06, 3072, 0x00, 16000, 0x0001, 1, 0x00, 0x00, 16, 1, 0x01, 16, 1,
0x00, 0x00, 16, 17, 0x01, 16, 17, 0x00, 0x00

//write air mode transparent.
HCI_Write_Voice_Setting 0x0063

HCI_Vs_Wbs_Associate 0xFD78, 0x1

HCI_Setup_Synchronous_Connection 1, 0x1f40, 0x1f40, 0xE, 0x0063, 0x02, 0x03c8
```

4. To disable WB speech, use this script:
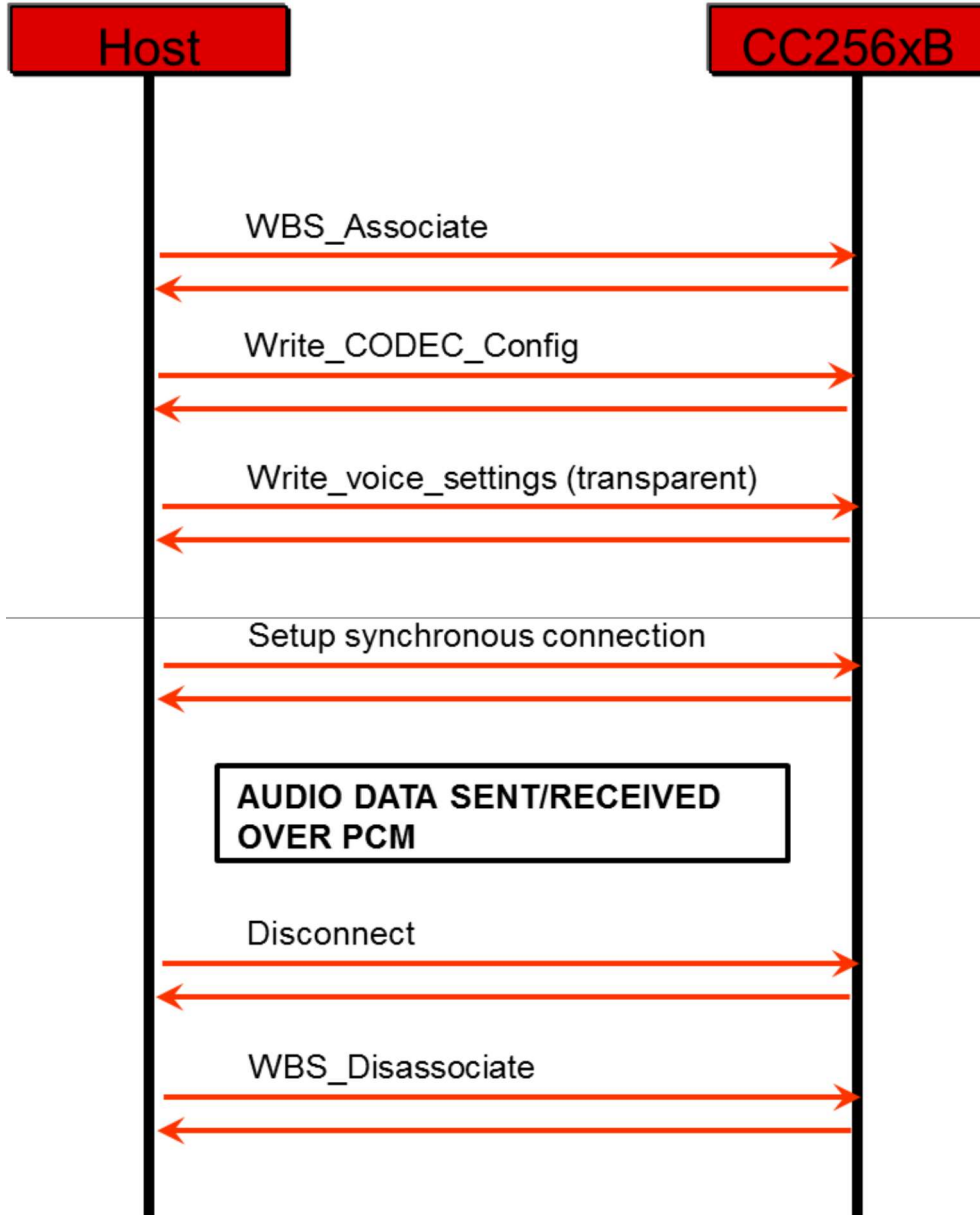
```
HCI_Disconnect (0x0406) 0x101, 0x13

HCI_Vs_Wbs_Disassociate 0xFD79
```

5. To re-enable BLE or ANT:

- Disable AVPR: Send_HCI_VS_AVPR_Enable 0xFD92, 0, 0, 0, 0
  - Re-load BLE or ANT add-on

WBS Enable/Disable Flow Sequence

The following diagram shows typical flow sequence of a host (such as MSP430) initiating a WB speech connection.
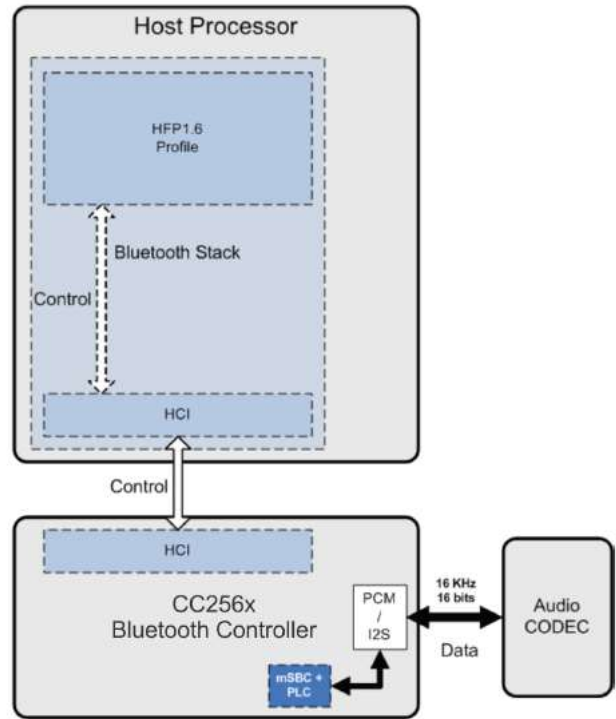

WB in NB (Legacy) and WB Speech Bluetooth Headsets

The device normally interfaces to the host at 16 Ksamples per second and performs the mSBC encode and decode processes. This configuration eliminates the need for WB speech SBC encoding on the host side as shown below.

HFP 1.6 Architecture
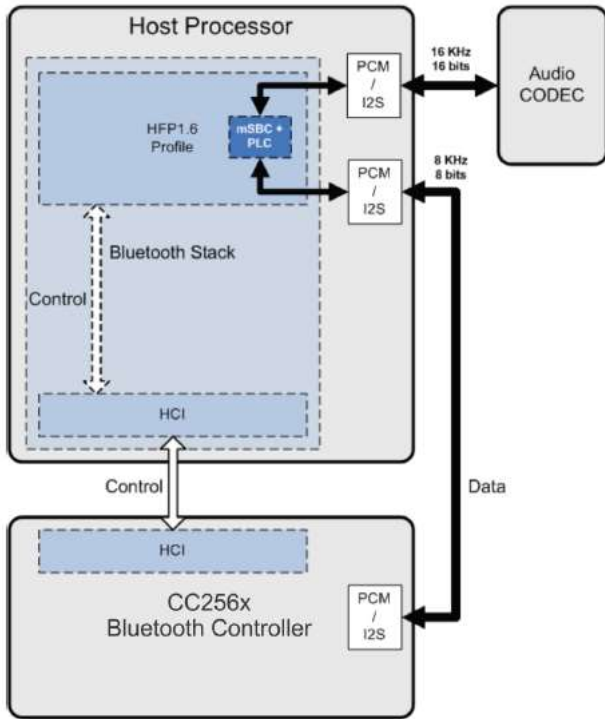
Assisted HFP 1.6 Architecture

As shown in the diagram above WBS allows 16KHz sampling rate at the same air speed of 64kbps.

Combinations of Host Input and Peer Device Type

CC256x PCM Bus (Determined by Host)	Headset Type of Bluetooth Peer	Bluetooth Air Interface	Comments
A-Law, U-Law or 8-kHz	NB	CVSD	Voice connection is NB
A-Law, U-Law or 8-kHz	WB	CVSD	Voice connection is NB
16-kHz linear	WB	mSBC	Voice connection is WB

From the chart above, if the host provides 16KHz SBC encoded data on the PCM bus and the peer device also supports WBS, it will use WB Speech for the voice connection.

💡**Note:** Standard Bluetooth is able to handle WBS without using the 65-nM advanced audio features:

• The host provides 8-kHz SBC encoded PCM bus

• Bluetooth sends eSCO transparent over the air

{{

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **CC256x Advanced Voice and Audio Features** here.

Keystone=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **CC256x Advanced Voice and Audio Features** here.

C2000=*For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio Features** *here.*

DaVinci=*For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio Features** *here.*

MSP430=*For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio Features** *here.*

OMAP35x=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio Features** *here.*

OMAPL1=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio Features** *here.*

MAVRK=*For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article* **CC256x Advanced Voice and Audio**

For technical s... please post you questions at http://e2e.ti.cor Please post on... comments abo... article **CC256x Advanced Voi... Audio Feature**

}}

Voice   and   Audio
Features here.

Features
here.

Links

Amplifiers & Linear	DLP & MEMS	Processors	Switches & Multiplexers
Audio	High-Reliability		Temperature Sensors & Control ICs
Broadband RF/IF & Digital Radio	Interface	- ARM Processors	Wireless Connectivity
Clocks & Timers	Logic	- Digital Signal Processors (DSP)	
Data Converters	Power Management	- Microcontrollers (MCU)	
		- OMAP Applications Processors	