

Processor SDK Automotive Audio Getting Started Guide



Last updated: 07/06/2017

Contents

Introduction

Setup software

- Code Composer Studio
- EVE compiler
- Processor SDK Automotive Audio

Setup EVM hardware

Setting up CCS for EVM and Processor-SDK Automotive Audio

Running examples/demonstrations

- Bare metal examples
- TI-RTOS kernel examples
- Peripheral driver examples
- EVE FFT/IFFT Demonstration
- EVE PFC Demonstration
- Multicore Streaming Demonstration

Next steps

How to

- Speed up downloading the installer

Archive

NOTE

Processor SDK Automotive Audio has been renamed to Processor SDK RTOS Automotive. This page is no longer being actively maintained. Please view [Processor SDK RTOS Automotive Getting Started Guide](#) for the latest Getting Started Guide

Introduction

The **Processor Software Development Kit (Processor-SDK)** provides the core foundation and building blocks that facilitate application software development on TI's embedded processors. This *Getting Started Guide* focuses on the Real-time operating system (RTOS) and provides information on getting the software and running basic examples/demonstrations bundled in the SDK.

By the end of this *Getting Started Guide*, the user will have

- Installed Code Composer Studio
- Installed latest Emulation package (for connecting to target using JTAG)
- Installed the Processor SDK Automotive Audio software
- Executed the RTOS out-of-box application

Setup software

See the [Release Notes](#) for information on minimum requirements for host.

Code Composer Studio



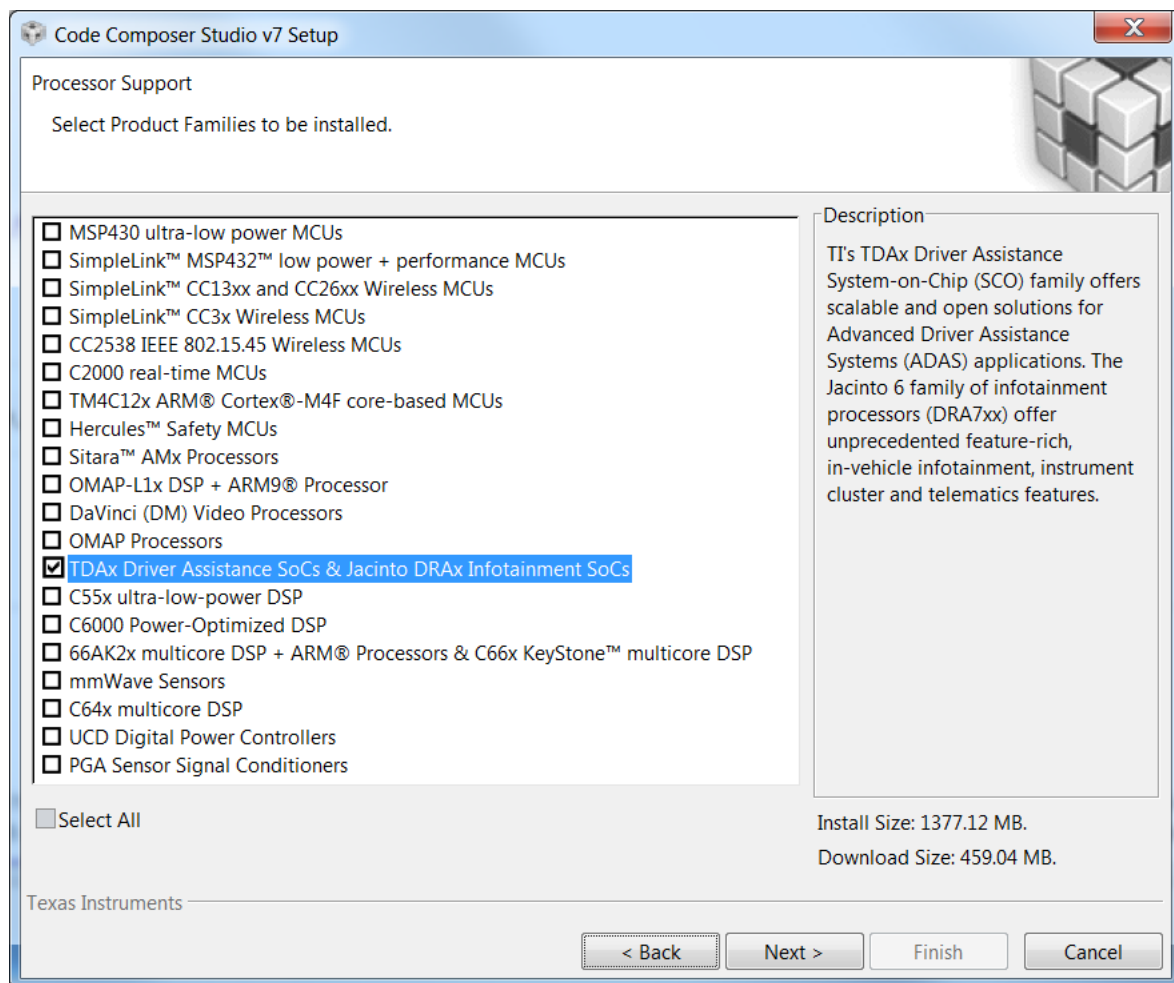
The Processor SDK Automotive Audio uses *Code Composer Studio* as the host integrated development environment for development and debug. To download CCS, see [this page \(http://processors.wiki.ti.com/index.php/Download_CCS\)](http://processors.wiki.ti.com/index.php/Download_CCS).

NOTE

See the [Release Notes](#) for the recommended version of CCS. This is the version that was validated with the software in the SDK and will provide the best user experience.

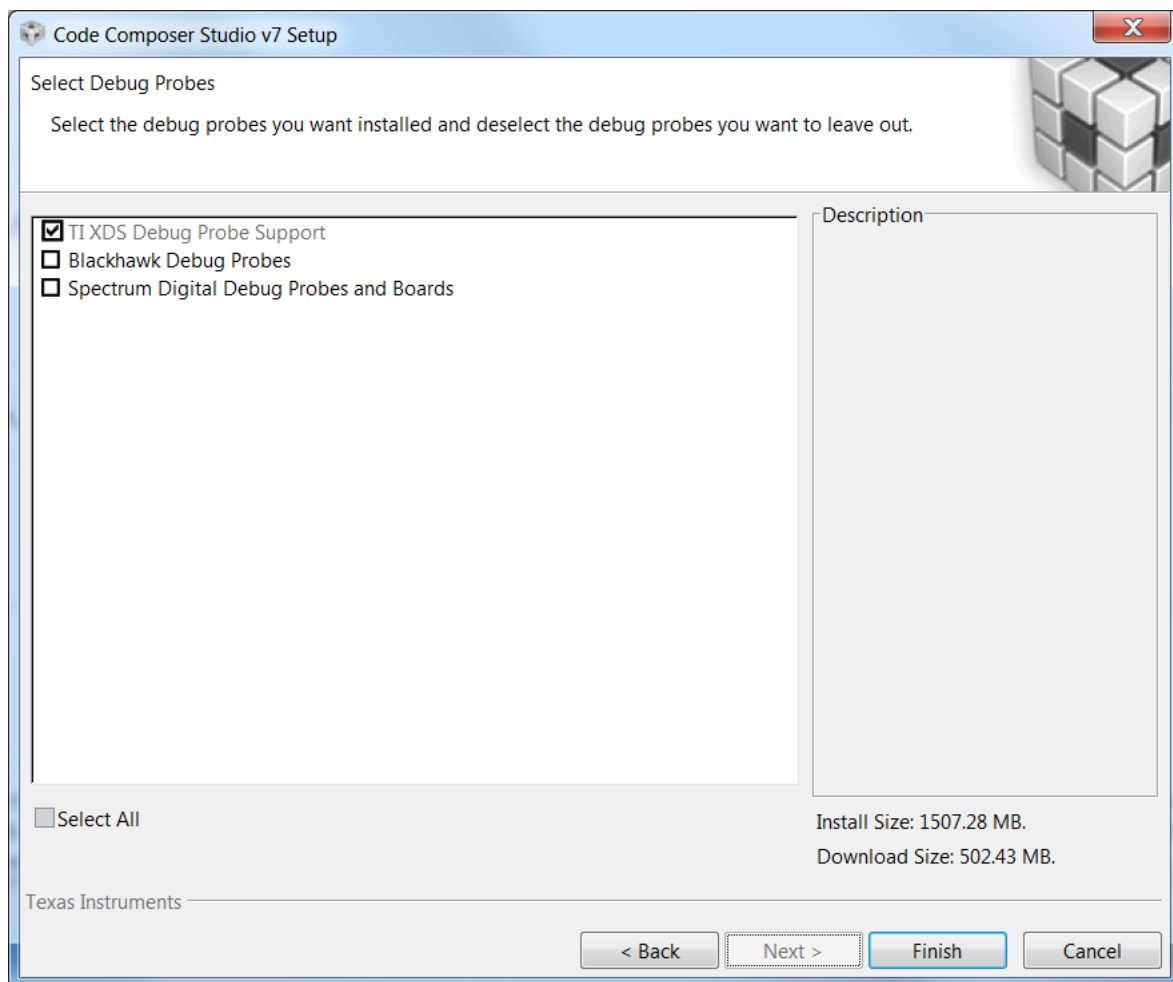
To install CCS, please refer to the instructions provided in the Code Composer Studio v7 landing page (http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v7).

When installing CCS, you can choose to control what is installed for processor architecture.



NOTE

If you plan to use emulators other than the XDS100 class or XDS200 class of emulators, please select the appropriate emulation drivers at the time of install. Code composer studio does not allow upgrades on drivers that were not installed during the first install.



EVE compiler

Selecting the EVE compiler add-on is no longer performed as part the CCS installation process. After CCS installation has completed, navigate to 'Help -> CCS App Center' and select the EVE Compiler to install as a CCS Add-on. Follow the dialog to complete the installation.

NOTE

If this step is skipped, PDK CSL builds will fail for DRA75x and DRA78x devices. Also the EVE Audio demos will not be available. This is not applicable for the DRA72x device.

Processor SDK Automotive Audio

The final step is to download and install the Processor SDK Automotive Audio. See the software product page to get the latest version of this software:

- Processor SDK for DRA7XX (<http://www.ti.com/tool/PROCESSOR-SDK-DRA7X>)

From the software product page, go to the download page by clicking "Get Software" for the RTOS Automotive package.

NOTE

- To avoid configuring CCS "Tool Discovery Path" to search for components in different directories, it is recommended to install the SDK in the same directory as CCS. This is `C:\TI` for Windows and `/home/[user]/ti` for Linux.
- Once installer has started, the Cancel button may not work properly.

Setup EVM hardware

CAUTION The EVM board is sensitive to electrostatic discharges (ESD). Use a grounding strap or other device to prevent damaging the board. Be sure to connect communication cables before applying power to any equipment.

The EVM provides the ability to utilize a variety of capabilities of the SoC. Follow instructions in the included *EVM Quick Start Guide* for information on hardware configuration and other pertinent information. This guide is included in the EVM kit and also available for download from the software download page for your particular device. The list of supported EVMs are provided in the [Release Notes](#).

If you connect to the EVM UART, use the following host configuration:

- **Baud Rate:** 115200
- **Data Bits:** 8
- **Parity:** None
- **Flow Control:** Off


Setting up CCS for EVM and Processor-SDK Automotive Audio

After the Processor SDK is installed, launch Code Composer Studio and make sure that the components inside Processor SDK are discovered by the CCS eclipse environment. If you installed the SDK in the same directory as CCS, this is as simple as starting CCS and it will auto-detect the newly installed components.

NOTE

If you installed the SDK and CCS in different paths, see the [Custom Installation Path](#) **How To** page that provides instructions to configure for a custom installation path.

The next step is to make a connection between CCS and your EVM (or *target*). If you need help with this step, see the [Setup CCS](#) **How To** page explaining this further.

**Useful Tip**

At this point, you should be able to connect to target using CCS. Do not proceed to below steps until this is complete.

Running examples/demonstrations

The SDK comes with some simple *examples* to get started using the software and IDE. The *demonstrations* contain a richer set of software and perform more complex features. If new to the SDK, it is suggested to go through the examples before writing your application.

Bare metal examples

The bare-metal examples gets user started with development of code without an operating system. These are simple examples that does not get into details of software components provided in the SDK.

TI-RTOS kernel examples

The RTOS examples gets user started with development of code using the TI-RTOS real-time operating system. These are simple examples that does not get into details of software components provided in the SDK.

Peripheral driver examples

For peripheral driver examples that are provided in Platform Development kit (PDK), please refer to [PDK Example and Test Project Creation](#) for generating CCS project for driver examples supported in the SDK.

EVE FFT/IFFT Demonstration

This demonstration is used to show how to offload a processing intensive operation to the Embedded Vector Engine (EVE), which can free up other processors to perform additional processing (i.e. DSP). The operation offloaded to the EVE vector core in this example is a 1024-point, 32-bit data, 32-bit twiddle factor, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). Refer to `processor_sdk_rtos_dra7xx_[version]/demos/eve_audio/examples/fft/readme.txt` for more information on building and running this example.

An application report "Getting Started with EVE Audio" is also included under `processor_sdk_rtos_dra7xx_[version]/demos/eve_audio/doc/DRA7XX_EVE_Audio` to more generally present the EVE programming environment and provide a guideline to start developing audio software for the EVE core.

EVE PFC Demonstration

This demonstration extends the EVE FFT/IFFT functionality by implementing a complex multiply kernel, optimized memory management scheme, and basic DSP host framework to perform Partitioned Fast Convolution operation on the EVE. The demo includes two sample 3000-tap low-pass FIR filters with 200 Hz and 2 kHz cutoff frequencies. The filters may be used as-is or modified to achieve the desired frequency response. Refer to `processor_sdk_rtos_dra7xx_[version]/demos/eve_audio/examples/pfc/readme.txt` for more information on building and running this example.

Multicore Streaming Demonstration

The multicore streaming demonstration uses IPC APIs to implement a streaming usecase using cached zero-copy memory between two cores in the system. Refer to `processor_sdk_rtos_dra7xx_[version]/demos/multicore_streaming/readme.txt` for more information on building, running, and validating the example.

Next steps

Now that you have a solid baseline set up, see the [Processor SDK RTOS Software Developer Guide](#) for more information.

The SBL (Boot) component in this SDK different than the component which is described in [Processor SDK RTOS Boot](#). Please refer to the [SBL_UserGuide.pdf](#) found under `pdk_dra7xx_[version]/packages/ti/boot/sbl_auto/doc`

Speed up downloading the installer

The size of the installer is large since we want to provide one bundle for all the components. The bad side of this is that if you are manually downloading the Processor-SDK installer, you may run into issues such as download stall or slow download. One simple solution is to run a download manager/accelerator. One open source solution is <http://www.freownloadmanager.org/>.

Processor SDK Automotive Audio 03.01.01 (http://processors.wiki.ti.com/index.php?title=Processor_SDK_Automotive_Audio_Getting_Started_Guide&oldid=223473)

keystone=

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 [Processor SDK Automotive Audio Getting Started Guide](#) 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 [Processor SDK Automotive Audio Getting Started Guide](#) here.

C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article [Processor SDK Automotive Audio Getting Started Guide](#) here.

Links

[Amplifiers & Linear](#)

[Audio](#)

[Broadband RF/IF & Digital Radio](#)

[Clocks & Timers](#)

[Data Converters](#)

[DLP & MEMS](#)

[High-Reliability](#)

[Interface](#)

[Logic](#)

[Power Management](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers](#)

[Temperature Sensors & Control ICs](#)

[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=Processor_SDK_Automotive_Audio_Getting_Started_Guide&oldid=229374"

This page was last edited on 6 July 2017, at 12:17.

Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.