CrossCore® Development Tools by Analog Devices

µCUSB Device™ Stack for CrossCore® Embedded Studio Rel.2.6.0 Release Notes

Contents

1 µC/USB Device™ Stack for CrossCore® Embedded Studio Rel. 2.6.0 Release Notes

Supported Processors

Support is provided, in all of the µC/USB Device™ products, for ADSP-BF52x, ADSP-BF54x, ADSP-BF60x, ADSP-BF707, ADSP-SC589 and ADSP-SC573.

Software Requirements

Tools

µC/USB Device™ Stack for CrossCore® Embedded Studio 2.6.0 requires the installation of CrossCore® Embedded Studio 2.6.0, available at http://www.analog.com/cces

What's New

ADSP-SC589 500MHz Core Clock Support

All the examples provided with the µC/USB Device™ Class products are modified to use the default CGU0 settings done in init_code which is 450 MHz Core Clock and 225 MHz system clock.

The C/USB Device™ Stack now also supports 500 MHz Core Clock for the ADSP-SC589 processor. If you are developing for 500MHz ADSP-SC589 processor, the core clock speed can be set to 500 MHz by modifying and rebuilding the preloads/initcodes. Refer to the README.txt in the appropriate preload/initcode project for an explanation of how to increase the processor speed, and an indication of what the changes entail. The preload and initcode projects are located in the <Installation Directory>\SHARC\ldr\init_code\SC58x_init folder.

2 µC/USB Device™ Stack for CrossCore® Embedded Studio Rel. 2.4.0 Release Notes

Supported Processors

Additional support is provided, in all of the µC/USB Device™ products, for ADSP-BF52x, ADSP-BF54x, ADSP-BF60x, ADSP-BF707, ADSP-SC589 and **ADSP-SC573**. New to the revision 2.4.0 is support for the **ASDP-SC573** processor.

Tools

µC/USB Device™ Stack for CrossCore® Embedded Studio 2.4.0 requires the installation of CrossCore® Embedded Studio 2.4.0, available at http://www.analog.com/cces

What's New

New Processor Support

The µC/USB Device™ Stack now supports the ADSP-SC573 processor. Following Class drivers are offered for this new processor:

- µC/USB Device™ Class CDC-ACM for CrossCore® Embedded Studio
- µC/USB Device™ Class HID for CrossCore® Embedded Studio
- µC/USB Device™ Class MSC for CrossCore® Embedded Studio
- µC/USB Device™ Class PHDC for CrossCore® Embedded Studio
- µC/USB Device™ Class Vendor for CrossCore® Embedded Studio
- µC/USB Device™ Class Audio for CrossCore® Embedded Studio (new)

3 µC/USB Device™ Stack for CrossCore® Embedded Studio Rel. 2.0.0 Release Notes

Supported Processors

Additional support is provided, in all of the µC/USB Device™products, for ADSP-SC589 and ADSP-BF707. ADSP-SC589 has two USB ports, USB0 (USB OTG) and USB1 (USB HS). Both USB0 and USB1 ports support the USB device mode.

Tools

µC/USB Device™ Stack for CrossCore® Embedded Studio 2.0.0 requires the installation of CrossCore® Embedded Studio 2.0.0. USB device driver available in CrossCore® Embedded Studio 2.0.0. is updated to support the ADSP-SC589 processor in CrossCore® Embedded Studio 2.0.0.

New µC/USB Device™ Stack and updated USB device driver

µC/USB Device™ Stack CrossCore® Embedded Studio 2.0.0 is updated to the latest version V4.04.01.

µC\USB Device Vendor Class product now provides the corrected WinUSB driver installer (WinUSB_composite.inf) for the Windows host PC for the USB Composite Devices. This driver will automatically get installed during the installation of the µC\USB Device Vendor Class product.

µC/USB Device™ Stack currently doesn't supports the associated interfaces, so now when you use the Micrium Echo Demo Application with both the Echo Sync and the Echo Async options enabled, two different "ADI Vendor Specific USB Composite Devices" will get installed. Please refer to the readme provided with the µC\USB Device Vendor Class demo examples for more information.

USB Device driver is updated to add the support of ADSP-SC589. Some issues related to the detection of USB MSC devices were fixed.

Knows Issues

- The USB stack supports the connection to any of the USB0 (USB OTG) or USB1 (USB HS) ports provided on the ADSP-SC589 EZ-Board, but connection to both the ports at the same time is not supported. However, you can switch between USB0 (USB OTG) and USB1(USB HS) ports while your application is running.

- Demo application provided in the USB Device CDC_ACM Class from µC/USB Device™ Stack version 4.04.01 is changed. When you upgrade from a lower release version to 2.0.0 release, the application files present in the existing demo examples will not be overwritten with the updated ones available with the 2.0.0 installation. Hence it is recommended to first remove the existing µC-USBD CDC-ACM add-in from your demo example and then add the µC-USBD CDC-ACM 2.0.0 add-in.

4 µC/USB Device™ Stack for CrossCore® Embedded Studio Rel. 1.1.0 Release Notes

Introduction

These release notes describe the 1.1.0 release of the µC/USB Device™ Stack for CrossCore® Embedded Studio, a collection of µC/USB Device™ add-in products for CrossCore® Embedded Studio:

- µC/USB Device™ Core for CrossCore® Embedded Studio
- µC/USB Device™ Class CDC-ACM for CrossCore® Embedded Studio
- µC/USB Device™ Class HID for CrossCore® Embedded Studio
- µC/USB Device™ Class MSC for CrossCore® Embedded Studio
- µC/USB Device™ Class PHDC for CrossCore® Embedded Studio
- µC/USB Device™ Class Vendor for CrossCore® Embedded Studio
- µC/USB Device™ Class Audio for CrossCore® Embedded Studio (new)

These products can be obtained from the ADI website: http://www.analog.com/ucusbdevice.

Supported Processors

New to revision 1.1.0 is support for the ASDP-BF70x, ADSP-BF518 and ADSP-214xx. The following table gives the full list of supported processors.

Processor families	Available Products
ADSP-BF52x, ADSP-BF54x, ADSP-BF60x and ADSP-BF70x	All
ADSP-BF518, ADSP-BF533 and ADSP-BF561, ADSP-214xx	Vendor Class and Core only

Hardware Requirements

- The ADSP-BF533 and ADSP-BF561 implementation requires the addition of the Blackfin USB-LAN EZ-Extender,available at http://www.analog.com/EX1-USB.
- The ADSP-BF518 and ADSP-214xx implementation requires the addition of the USB EZ-Extender for Blackfin and SHARC EZ-Boards/EZ-KIT Lites, available at http://www.analog.com/EX2-USB.

- Use of the Audio Class SSM2603 codec interface driver requires a ADSP-BF526 EZ-Board or ADSP-BF527 EZ-KIT Lite development board.
- Use of the Audio Class ADAU1761 codec interface driver requires a BF609 EZ-KIT Lite or BF707 EZ-Board development board and the Audio EI3 Extender Board.

Please note that it is highly recommended that the latest versions of these extenders are used.

Software Requirements

Tools

- µC/USB Device™ add-ins for CrossCore® Embedded Studio 1.1.0 require the installation of CrossCore® Embedded Studio 1.1.0, available at http://www.analog.com/cces.
- µC/USB Device™ add-ins for CrossCore® Embedded Studio 1.1.0 require the installation of the C/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio 1.1.0, available at http://www.analog.com/ucos3.
- The ADSP-BF533 and ADSP-BF561 implementations require the addition of the Blackfin USB-LAN EZ-Extender Board Support Package 1.1.0 , available at http://www.analog.com/EX1-USBLAN.
- The ADSP-BF518 and ADSP-214xx implementations require the addition of the Blackfin/SHARC USB Extender Board Support Package 1.1.0, available at http://www.analog.com/EX2-USB.
- The µC/USB Device™ Class MSC for CrossCore® Embedded Studio 1.1.0 product requires C/FS™ File System for CrossCore® Embedded Studio 1.1.0 (if C/FS Storage layer is used with the MSC class), available at http://www.analog.com/ucfs.
- Use of the Audio Class ADAU1761 codec interface driver requires the Audio EI3 Extender Board Board Support Package release 1.1.0.

IMPORTANT NOTE: the use of any µC/USB Device class add-in products requires the addition of the µC/USB Device™ Core for CrossCore® Embedded Studio 1.1.0 product, available at http://www.analog.com/ucusbdevice.

What's New

Updated Micrium Release

All µC/USB Device™ add-in products have been updated to use the 4.02.00 release of the Micrium C/USB Device™ Stack. Please refer to the Micrium Release Notes for further details.

Documentation

Micrium have moved their documention on-line. it can be viewed from https://doc.micrium.com/display/DOCV40200/uC-USB-Device+Documentation+V4.02.00. As a result, it is no longer displayed in the CrossCore® Embedded Studio help.

New classes

New to µC/USB Device™ Stack for CrossCore® Embedded Studio 1.1.0:

µC/USB Device™ Class Audio for CrossCore® Embedded Studio - The Audio class allows you to stream audio to and from a host PC. The product comprises four types of codec interface:

Codec Interface Add-in	Description	Processor Family
uC-USBD Audio (Simulated Codec) for ADSP-BF52x /BF4x uC-USBD Audio (Simulated Codec) for ADSP-BF60x /BF70x	Enables either simulated record or record and playback to demonstrate USB Audio.	ADSP-BF52x, ADSP-BF54x, ADSP-BF60x and ADSP-BF70x
uC-USBD Audio (SSM2603 Codec) for ADSP-BF52x	Provides record and playback of 16-bit Stereo PCM at 48kHZ using the SSM2603 codec.	ADSP-BF52x
uC-USBD Audio (ADAU1761 Codec) for ADSP-BF60x /BF70x	Provides record and playback of 16-bit Stereo PCM at 48kHZ using the ADAU1761 codec.	ADSP-BF60x and ADSP-BF70x
uC-USBD Audio (Template Codec) for ADSP-BF52x /BF54x uC-USBD Audio (Template Codec) for ADSP-BF60x /BF70x	Blank template to enable you to interface to your chosen codec.	ADSP-BF52x, ADSP-BF54x, ADSP-BF60x and ADSP-BF70x

Out of the box there is a simulated codec example which can be modified to exchange the simulated codec add-in with either the SSM603 or ADAU1761 codec add-in. For these the hardware and software requirements are given in the table below.

Codec	Development Board	Extender board	Board Support Package
SSM2603	ADSP-BF526 EZ-Board ADSP-BF527 EZ-KIT Lite	None required	CCES Driver library
ADAU1761	BF609 EZ-KIT Lite BF707 EZ-Board	Audio EI3 Extender Board	Audio EI3 Extender Board 1.1.0

Improvements

Buffer Allocation and Alignment

The Micrium 4.02.00 release dynamically allocates all internal buffers, no matter how large or small, and ensures they are aligned according to the USBD_CFG_BUF_ALIGN_OCTETS macro defined in the µC/Application vertical tab of the µC/USB Device horizontal tab in the System Configuration Editor. The memory pool size is assigned in the µC/LIB tab.

As a result of this change, a call to the µC/LIB Mem_Init() function needs to be added to existing main() functions, ahead of the call to OSStart(). In addition, please ensure that there is sufficient memory assigned to the µC/LIB pool, which can be set in the "Memory allocator heap size (bytes)" field of the µC/LIB tab in the System Configuration Editor. The default of 8192 should be sufficient for most cases.

Class Improvements

µC/USB Device™ Class MSC for CrossCore® Embedded Studio

The following improvements have been made by Micrium :

• Added support for multiple logical units.

• Added support for host computer right-click eject option.

• Added support for removable media (e.g. SD card) insertion/removal when µC/FS is used with MSC.

• Added support for high-level lock system ensuring an exclusive access to media shared between the host computer and an embedded file system application.

The single RAMDisk add-in, µC-USB Device Mass Storage Class, is replaced with three addins:

Add-in	Description
uC-USBD MSC (RAMDisk) for ADSP-BF52x/BF4x uC-USBD MSC (RAMDisk) for ADSP-BF60x/BF70x	This is the replacement add-in for the previous uC/USB Device Mass Storage Class add-in.
uC-USBD MSC (uC-FS) for ADSP-BF52x/BF4x uC-USBD MSC (uC-FS) for ADSP-BF60x/BF70x	This add-in adds the µC/FS Storage layer to enable access to other mass storage. You will need µC/FS™ File System for CrossCore® Embedded Studio 1.1.0 in order to use this option. The available mass storage types will depend on the target processor*. µC/FS™ File System 1.1.0 is available at http://www.analog.com/ucfs.
uC-USBD MSC (Template) for ADSP-BF52x/BF4x uC-USBD MSC (Template) for ADSP-BF60x/BF70x	This add-in provides a blank Storage Layer to enable you to interface to the mass storage solution of your choice.

* Please note that the **uC-FS Mass Storage for Blackfin** add-in, which is available with the µC/FS™ File System for CrossCore® Embedded Studio 1.1.0 product, is only applicable for use with the µC/USB Host stack and as such cannot be used with the **uC-USBD MSC (uC-FS)** add-in. Otherwise the choice of storage media will depend on the processor target. Please refer to the µC/FS™ File System Release Notes for further information.

To convert the MSC example, e.g. **MSC Demo for ADSP-BF707**, from using the **uC-USBD MSC (RAMDisk) for ADSP-BF60x/BF70x** add-in to using the **uC-USBD MSC (uC-FS) for ADSP-BF60x/BF70x** add-in:

1. Remove the **uC-USBD MSC (RAMDisk)** add-in.

1. Add both the **uC-USBD MSC (uC-FS) for ADSP-BF60x/BF70x** and the appropriate µC/FS™ File System add-in: see above table.

1. Replace the **USB Examples' Support** add-in with the **uCOS-III for Blackfin** add-in as described in the example readme.html file.
2. Edit the System/uC-USB/Device/App/app_usbd_msc.c file to enter the appropriate µC/FS name by changing the definition of _DEVICE_NAME_ to be the string used in the call to FSDev_Open (see System/uC-FS/fs_app.c for details of these names).

Change configuration settings as required in accordance with the µC/FS documentation. Please refer to the µC/FS™ File System for CrossCore® Embedded Studio 1.1.0 release notes for documentation details.

User Interface Settings

The following products have modified content in the System Configuration Editor. This is to accommodate new settings introduced with version 4.02.00 of the Micrium µC/USB Device™ Stack, and to improve the user experience.

1. **Core:** New Micrium settings.
2. **Vendor:** New Micrium settings; Better control over how app_usbd_vendor.c can be configured, to enable the original Micrium example to be used or to use the ADI specific example.
3. **MSC:** New Micrium settings and support for removable media presence detection for the uC-USBD MSC (uC-FS) add-in only.

Example Browser

Components have been renamed from a generic name (e.g. **uC-USB Device Vendor Class**) for all processor targets to names that include the target processors (e.g. **UC_USBD Vendor Class for ADSP-BF60x/70x**). This makes it easier to navigate the Example Browser as the Add-in drop-down list now displays a unique set of add-ins, enabling you to narrow down the Search results upon selection of the required Add-in. Please note, however, that this list is derived from the full set of add-ins included in the selected Product and hence some Add-in items will result in no Search Results since examples do not exist for all Add-ins and processor targets. For example, MSC examples are only available for the uC-USBD MSC (RAMDISK) add-ins. Another example is the Vendor Class product where examples are not available for ADSP-BF533, ADSP-BF518, ADSP-BF561 and SHARC 214xx; Examples for these processor targets are provided with the appropriate USB extender BSP detailed above in the **Software Requirements** section.

Examples

The insertion of the µC/OS-III libraries that are linked in to each of the examples has been simplified. These libraries are now inserted via the **USB Examples' Support** add-in that is supplied with the µC/USB Device™ Core for CrossCore® Embedded Studio 1.1.0 product. The use of an add-in simplifies the replacement of the µC/OS-III libraries with the µC/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio 1.1.0 product. Please note, you are allowed to use the given example with the **USB Examples' Support** add-in

and extend it for exploratory purposes. However, should you wish to extend the application to meet your own product requirements for redistribution you will be required to replace it with an appropriate RTOS. We recommend that you obtain the full µC/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio release 1.1.0, available from http://www.analog.com/ucos3.
The readme.html file for each example details what is required to replace the USB Examples' Support add-in with the µC/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio add-in.
To run the examples the following development boards are required:

Target	Development Board	Part Number	Board Rev	BOM Rev	Silicon Rev
ADSP-BF526	ADSP-BF526 EZ-Board	ADZS-BF526-EZBRD	1.1	1.9	0.2
ADSP-BF527	ADSP-BF527 EZ-KIT Lite	ADZS-BF527-EZLITE	2.2	3.3	0.2
ADSP-BF548	ADSP-BF548 EZ-KIT Lite	ADZS-BF548-EZLITE	1.4	2.5	0.4
ADSP-BF609	ADSP-BF609 EZ-KIT Lite	ADZS-BF609-EZ-BRD	1.0	1.4	0.2
ADSP-BF707	ADSP-BF707 EZ-Board	ADZS-BF707-EZBRD	1.0	1.4	0.0B

While the µC/USB Device™ Class Vendor product has add-in support for development boards that make use of either the Blackfin USB-LAN EZ-Extender (http://www.analog.com/EX1-USB) or the USB EZ-Extender for Blackfin and SHARC EZ-Boards/EZ-KIT Lites (http://www.analog.com/EX2-USB), examples are not provided with the Vendor product itself; Instead, these examples are distributed with the appropriate Board Support Package as detailed in the Software Requirements section above.

Known Issues

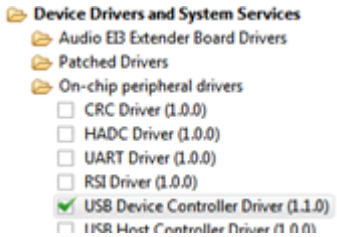
USBD Audio Class Settings required for ADSP-BF707 and the ADSP-BF609

If you are using the USBD Audio Class on the ADSP-BF70x or ADSP-BF60x processor and you discover enumeration and/or sound quality issues then the USB driver needs to

be modified. The driver will no longer be using FIFO double-buffering. In addition, for ADSP-BF70x release mode builds the USB driver must be added to your project and compiler optimization must be disabled. The procedure to modify the driver and settings is as follows.

1. Add the USBD driver to your project:

- Open the "System Configuration Editor and navigate to the “System Configuration Overview" page;
- Click on the "Add ..." button and select the "USB Device Controller Driver (1.1.0)”:



2. Modify the driver:

- In the "Project Explorer" window navigate to the usb driver source file which is now located at "system" > "drivers" > "usb" > "controller" > "musbmhdrc" > adi_usbd_musbmhdrc.c;
- Double click on "adi_usbd_musbmhdrc.c" which will open the source file in the editor window;
- Scroll down to line 34 "#include "adi_usb_dev_musbmhdrc.c";
- Right click on the line and choose the menu item "Open Declaration" which will open the file in the editor;
- In the file "adi_usb_dev_musbmhdrc.c" scroll down to line 2430 which should read "adi_usbdrv_SetDynamicFifoParams(pUsbDrvData,pEpInfo,true);";
- Change the value of "true" to "false".

3. For Release Mode builds disable compiler optimization for the USB driver on the ADSP-BF70x

- In the "Project Explorer" window navigate to the usb driver source file which is now located at "system" > "drivers" > "usb" > "controller" > "musbmhdrc" > adi_usbd_musbmhdrc.c
- Right click on "adi_usbd_musbmhdrc.c" and select “properties” which will bring up a properties dialog box.
- Select “C/C++ Build > Settings” and then in the “Tool Settings” tab select “CrossCore Blackfin C/C++ Compiler > General”
- Finally, deselect “Enable Optimization”

4.1 Previous releases of µC/USB Device™ Stack for CrossCore® Embedded Studio are not compatible with CCES 1.1.0

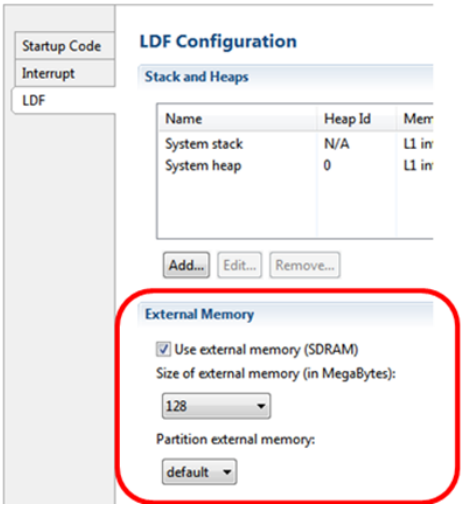
CrossCore® Embedded Studio1.1.0 contains a new integrated (device and host mode) USB Controller device driver, that has resulted in a new API between the Micrium USB Device stack and the device driver. For this reason it is not possible to use releases 1.0.0 or 1.0.1 of µC/USB Device™ Core for CrossCore® Embedded Studio with CCES 1.1.0.

4.2 New ADSP-BF609 Projects with both the µC/USB Device and the Startup Code/LDF add-ins fail to build

Creating a New CrossCore Project for ADSP-BF609 which adds both µC/USB Device and the Startup Code/LDF add-ins at the same time will fail to build with the message,

```
[Error pp0019] "..\system\startup_ldf\app.ldf":209 'There is no support for non-par
```

To work round the problem only add the Startup Code/LDF add-in after the µC/USB Device add-ins are added. Once added in this way, the Startup Code/LDF add-in must be configured to enable the use of external memory with the default settings:



Adding µC/OS-III add-in to Project with µC/USB Device add-ins already included
When adding the µC/OS-III add-in to project at the same time as adding the µC/USB Device core add-in, the Add-in Framework will apply the necessary settings to correctly configure µC/OS-III. This is also the case when the µC/USB Device Core add-in is added to a project already containing the µC/OS-III add-in. However, it is not possible to apply these settings when the µC/OS-III add-in is added to a project that already contains the µC

/USB Device core add-in. You can find out the settings to apply, in this last case, from the example readme.html file that is relevant to your target processor. This information is contained in section 2.2 of the appropriate readme.html file which describes the replacement of the USB Examples' Support add-in.

Incorrect/Incomplete instructions in µC/USB Device™ Stack 1.0.1 examples to replace RTOS libraries with µC/OS-III Add-in

When following the instructions in the readme.html files of examples for the µC/USB Device™ Stack 1.0.1 release, please note that the instructions should be applied to all build configurations ([All configurations]). Please also note that in the readme.html file for the MSC Product 1.0.1 examples, the libraries, libuccpu.dlb and libuclib.dlb are listed in error: Entries in C/C++ Build => Settings => CrossCore Blackfin Linker => "Additional Options" do not exist and so do not need to be removed.

Replacing USB Examples' Support add-in with µC/OS-III Add-in

The binaries installed with the USB Examples' Support add-in have been built in release configuration. It has been found that when the µC/OS-III sources are built in debug configuration the PHDC example will fail to run properly on the ADSP-BF526 & ADSP-BF527 targets. This is due to adverse interaction between the Timer and Statistics task. To resolve the problem, adjust the Task Rate of the Statistics task to 20 instead of 200 as described in the example readme. Please note that the instructions in the readme.html files of examples should be applied to all build configurations ([All configurations]).

Cannot Use ROM versions of uC/OS-III for ADSP_BF707

The µC/USB Device™ product add-ins cannot be used with the uCOS-III ADSP-BF70x ROM Configurations 1, or 2. The RAM variant, uCOS-III for Blackfin add-in must be used instead.

µC/USB Device™ Stack can only be added to one Core of Dual-Core Targets

The µC/USB Device™ stack can only be executed from a single core of dual-core targets such as ADSP-BF561 & ADSP-BF60x.

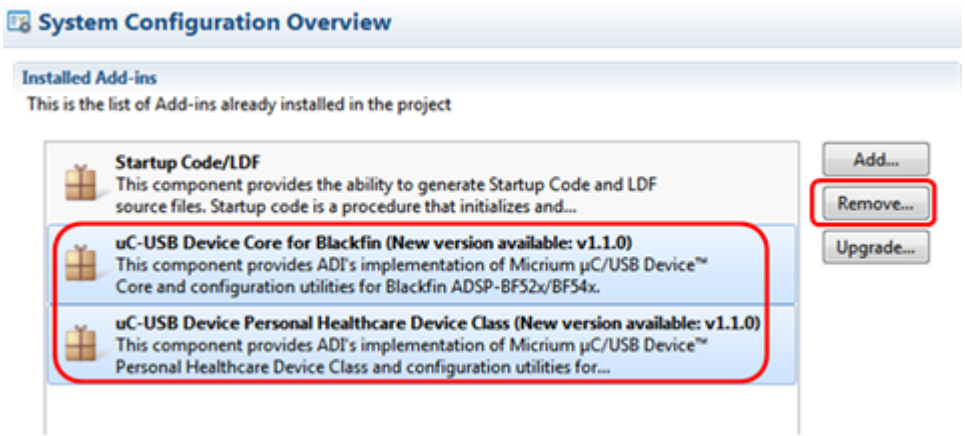
Upgrading Projects from Previous Releases

It is not possible to apply the Add-in upgrade process to your application based on either the 1.0.0 or the 1.0.1 releases.
However, you can take the following steps to upgrade from the µC/USB Device 1.0.1 add-ins to the µC/USB Device 1.1.0 add-ins.

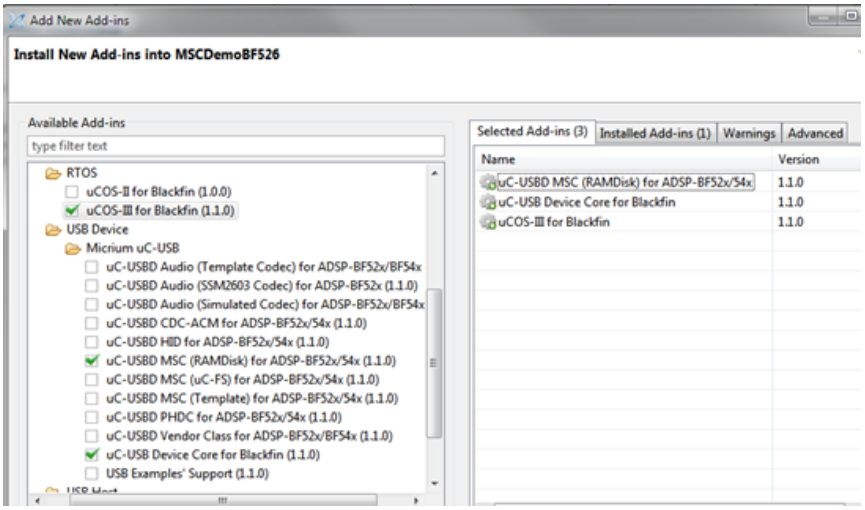
1. Make a backup copy of your existing project. In particular, you need to save aside any app_usbd_<class>.c files that you have modified, .e.g. app_usbd_vendor.c.
2. Note down all the settings (µC/USB Device, µC/LIB, µC/CPU, µC/OS-III) that you have changed via the System Configuration Editor.
3. Close CrossCore Embedded Studio (CCES) and install the latest µC/USB Device products as per your requirements.

4. Restart CCES and open the project you wish to upgrade
5. Double-click on the Project system.svc file to open the System Configuration Editor and click on the Overview tab to display the list of add-ins. If newer versions are available they will be indicated in red.

6. If your Project is an extension to an example, you should follow the instructions in the readme.html to replace the µC/OS-III library binaries with the µC/OS-III 1.1.0 add-in. Please ensure that you select [All configurations]. The µC/OS-III add-in should be added at the same time as adding the µC/USB Device 1.1.0 add-ins in step 9, below.
7. Remove both the uC-USB Device Core and the appropriate class add-ins by selecting the add-ins using the Ctrl key to select all simultaneously, and clicking the Remove... button.



8. You will now be presented with a summary of the remove operation. Ignore any warnings and proceed to remove the add-ins.
9. Next, add the new versions of the add-ins, by selecting all simultaneously:



10. Reapply the settings you saved aside in step 2.
11. Add a call to Mem_Init() in main() after the call to BSP_Init().
12. Ensure that you either replace or amend the new app_usbd_<class>.c file to include any modifications that you made.
13. If you are upgrading an extension to the VendorDemo application you will also need to change the following, if required by your extended application:
    a. The location of src/adi_stdio_ucusb.c. If not required it can be deleted from the project, along with any code that accesses the functions in it. To upgrade the link to the file, please do the following:
        i. Select the file int the Project Explorer and select the right-click -> Properties menu option.
        ii. Edit the Resource Location value to change COM_ANALOG_MICRIUM_UCUSBD_VENDOR_BASE_1_0_1_LOC to COM_ANALOG_MICRIUM_UCUSBD_VENDOR_BASE_1_1_0_LOC.
    b. The include path to hostapp.h: Open the Project Settings and modify the additional include directories in both the Assembler and Compiler sections for all configurations to change "${COM_ANALOG_MICRIUM_UCUSBD_VENDOR_BASE_1_0_1_LOC}/Host/Examples/Windows/VendorBulk" to "${COM_ANALOG_MICRIUM_UCUSBD_VENDOR_BASE_1_1_0_LOC}/Host/Examples/Windows/VendorBulk/Visual Studio 2010".
    c. Replace the call to App_USBD_BulkTestApp() in the App_USBD_Vendor_ADIBulkTask() function of app_usbd_vendor.c

µC/USB Device 1.0.1 projects fail to build if µC/FS 1.0.2 or µC/USB Host 1.0.0 also installed

Due to changes in the latest Micrium common components, µC/LIB and µC/CPU, included with the µC/FS 1.0.2 and µC/USB Host 1.0.0 products, applications that use any of these with the µC/USB Device 1.0.1 product will not build, as visibility of settings in app_ucusbd_cfg.h is lost resulting in several compilation errors e.g.:

```
"C:\Analog Devices\uCUSB_Device_Core-Rel1.1.0\uC-USBD\Blackfin\Device\ADI/musbhdrc/

  ADI_USBD_CFG_VENDOR_ID, /* Vendor ID. */
```

To work around the issue, please edit the system/uC-USB/Device/Ports/Blackfin/Device /usbd_dev_cfg.c file of your µC/USB device project to add the following line at top of the file:

#include <app_cfg.h>

This problem is fixed with the µC/USB Device™ Stack for CrossCore® Embedded Studio 1.1.0 products.

Errata

In the 1.0.0 release notes, the section, Adding µC/USB Device Stack to a project, incorrectly stated that without the selection of an RTOS add-in you would "not be able to proceed unless you also select uCOS-III for Blackfin (1.0.0) as an additional Add-In". In fact, with that release and with all subsequent releases, you are able to proceed with the addition of the µC/USB Device add-ins; the warning is simply a reminder that you need to satisfy the requirement before a viable application can be built.

5 µ C/USB Device™ Stack for CrossCore® Embedded Studio Release 1.0.1 Release Notes

5.1 Supported Processors

Support is provided, in all of the µC/USB Device™products, for ADSP-BF52x, ADSP-BF54x, ADSP-BF60x. Support for ADSP-BF533 and ADSP-BF561 is only provided in µC /USB Device™ Class Vendor for CrossCore® Embedded Studio.

5.2 Hardware Requirements

The ADSP-BF533 and ADSP-BF561 implementation requires the addition of the Blackfin USB-LAN EZ-Extender rev 1.1 (part number: ADDS-USBLAN-EZEXT) or above.

5.3 Software Requirements

5.3.1 Tools

µC/USB Device™ Stack for CrossCore® Embedded Studio1.0.1 requires the installation of CrossCore® Embedded Studio 1.0.1.2.

Software Products

µC/USB Device™ Stack for CrossCore® Embedded Studio requires support for uC/OS-III. All of the examples ship with a pre-configured version of µC/OS-III™ Real Time Kernel for CrossCore® Embedded Studio 1.0.0, provided as a binary library that is linked into the examples. The exact configuration of the RTOS is provided in the example documentation.

When multiple µC/USB Device Products are used in the same project, only µC/USB Device products with the same revision number can be used. Please note all µC/USB Device Class products require the µC/USB Device Core product.

The RTOS library can be replaced with either release 1.0.0 or 1.0.1 of µC/OS-III™ Real Time Kernel for CrossCore® Embedded Studio, as detailed in the example documentation.

Board Support Packages (BSP)

The ADSP-BF533 and ADSP-BF561 implementation requires the installation of the Blackfin USB-LAN EZ-Extender BSP revision 1.0.1 or higher.

5.4 Examples

Examples are available for ADSP-BF526, ADSP-BF527, ADSP-BF548, ADSP-BF533 and ADSP-BF561. In the case of ADSP-BF533 and ADSP-BF561, examples are provided with the µC/USB Device™ Class Vendor product. Examples can be built for both Debug and Release configurations.

All examples now copy to the workspace, when opened from the Examples Browser. Please note that, due to the way in which the CrossCore® Embedded Studio IDDE generates the path, it is common to exceed the Windows 255 character path limit. You are strongly advised to use a workspace other than the default for CCES, and with as short a path as possible, (e.g. C:\WORKSPACE-1\).

The µC/USB Device™ Class Vendor product examples have been extended to add support for the Micrium supplied example (StressTest). The example can either be built as the original (release 1.0.0) bulk loopback example or as the Micrium stress test example. An additional host application is supplied to use with the stress test example. Please refer to the example documentation for more information.

6 µC/USB Device™ Stack for CrossCore® Embedded Studio Rel. 1.0.0 Release Notes

What is µC/USB Device™ Stack for CrossCore® Embedded Studio

µC/USB Device™ Stack for CrossCore® Embedded Studio is the result of a partnership between Analog Devices and Micrium to provide a user-friendly programming environment for embedded applications that require USB device functionality. It is designed for embedded systems with USB device controllers such as provided with the Analog Devices' ADSP-BF60x processor family. Use of the stack requires the C /USB Device™ Core for CrossCore® Embedded Studio product along with one or more supported device class products described below.

The µC/USB Device™ Stack is provided with a hardware abstraction layer and it is modified to easily work with any USB device controllers supported by Analog Devices' processor family. Drivers for several common device classes (Mass Storage, CDC-ACM, HID, PHDC) are offered. A framework for developing vendor-specific class drivers is also provided.

The µC/USB-Device™ Stack uses a modular architecture with three software layers between the application and the hardware:

- The Device Controller layer interfaces with the USB Controller driver to process interrupts, notify the device core of bus events, and to receive/transmit packets.
- The Device Core layer controls packet reception and transmission, and responds to standard host requests during enumeration (the process by which a host determines the features of a device).
- The Class layer provides functionality to the host using one or more class drivers. Each class driver responds to class-specific requests and may provide an API for controlling some features and receiving/transmitting information.

Class Support

µC/USB Device™ Class CDC-ACM for CrossCore® Embedded Studio - The Communication Device Class (CDC) encompasses several communication models. Typical applications include modems, telephone systems and fax machines. The Abstract Control Model (ACM) converts the USB device into a serial communication device, and the target is recognized by the host as a serial interface (USB2COM, Virtual COM port).

µC/USB Device™ Class HID for CrossCore® Embedded Studio - The Human Interface Device (HID) Class allows you to implement any kind of user-input device. It can also be used to communicate with the host (without a special host driver) using a vendor-specific communication protocol. Typical applications include mouse, keyboard, game pad, etc....

µC/USB Device™ Class MSC for CrossCore® Embedded Studio - The Mass Storage Class (MSC) allows you to use the embedded target device as a USB mass storage device. Typical applications include USB memory stick, digital camera, MP3 player, DVD player, etc....

µC/USB Device™ Class PHDC for CrossCore® Embedded Studio - The Personal Healthcare Device Class (PHDC) allows you to set up the embedded target as a personal healthcare device, which can use a vendor-defined or IEEE-11073 based protocol. Typical applications include glucose meter, blood pressure monitor, weighing scale, etc....

µC/USB Device™ Class Vendor for CrossCore® Embedded Studio - The Vendor class allows you to develop a custom vendor-specific device (at the application level). This class interacts with WinUSB on Microsoft Windows using a combination of Control, Interrupt and Bulk transfers. A Vendor Specific host application is required to communicate with the device.

6.1 Getting Started with µC/USB Device ™ Stack for CrossCore ® Embedded Studio

Installation

CrossCore® Embedded Studio v.1.0.0 or newer must be installed prior to installing any of the µC/USB Device™ Stack for CrossCore® Embedded Studio products. In addition, µC /USB Device™ Stack operation requires the support of an RTOS. µC/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio is a separate product that may be purchased and installed in support of µC/USB Device Stack. All of the examples that are provided in the various µC/USB Device Stack products require µC/OS-III support. The examples, however, make use of a pre-configured µC/OS-III library, for a quick out-of-the-box experience.

As previously outlined, there are three software layers involved and each is delivered in the products as outlined below:

- The USB Controller driver is delivered in source and binary form as part of the CrossCore Embedded Studio product. All of the remaining products leverage the binary form of the driver. The sources, though not normally needed, are available.

- The Device Core layer product is licensed as a stand alone product and must be installed before any of the Class layer products can be used. It is recommended, though not required, that the Device Core product be installed before installing any of the Class layer products.
- The Class layer products are licensed and installed as individual products.

Software Layer	Product	Notes
Layer 1	USB Contoller Driver	Binary Form as part of libdrv.dlb in: • $CCES_INSTALL\Blackfin\lib\bf609_rev_any • $CCES_INSTALL\Blackfin\lib\bf609_rev_none Source Form: $CCES_INSTALL\Blackfin\lib\src\drivers\source\usb
Layer 2	USB Device Core	
Layer 3	USB Device Class Drivers	
	µC/USB Device™ Class CDC-ACM	
	µC/USB Device™ Class Vendor	
	µC/USB Device™ Class PHDC	
	µC/USB Device™ Class MSC	

	µC/USB Device™ Class HID	

Please make sure to close CrossCore Embedded Studio before proceeding with the installation. If CrossCore Embedded Studio is left open during the installation, it will have to be restarted after installing the µC/USB Device Stack products in order for the changes to take effect, and for µC/USB Device to be available.

Both the µC/USB Device Stack and uC/OS-III install the following common products:

- µC/LIB . This software is always installed into Common Program Files directory. This location is determined by the %CommonProgramFiles(x86)% environment variable in 64-bit operating systems or by %CommonProgramFiles% in 32-bit operating systems.
- µC/CPU. This software is always installed into Common Program Files directory. This location is determined by the %CommonProgramFiles(x86)% environment variable in 64-bit operating systems or by %CommonProgramFiles% in 32-bit operating systems.

The default location for installation is under C:\Analog Devices, e.g. C:\Analog Devices\uCUSB_Device_Core-Rel1.0.0. Should you wish to use a different location, Analog Devices strongly strongly recommends installing the µC/USB Device Stack products outside of the Program Files directory to prevent possible permission issues related to UAC (User Access Control). If you have already installed the product under Program Files then we recommend that you uninstall it and re-install it in a different location.

Note: Multiple versions of the µC/USB Device Stack can be installed on the same system. Only a single instance of a specific version of the product can be installed on a system.

License Checking

The installation process checks for a separate license for each of the µC/USB Device Stack products. If a valid license is not detected, the installer will start the Manage Licenses utility for entering and activating a license. The installer will fail in a non-interactive mode when valid license is not present.

Installation Logging

The installer does not create a log file by default. If you encounter installation issues, you can generate an installation log file by running the installer from the command prompt.

Change to the directory containing downloaded installer executable and run the following from the command prompt to install the Device Core layer product:

ADI_uCUSB_Device_Core-Rel1.0.0.exe /v"/l*v c:\temp\installer.log"

Similarly, the Class layer products may also be installed from the command line as follows

Class Layer Product	Command Line Executable Name
µC/USB Device™ Class CDC-ACM	ADI_uCUSBD_Class_CDC_ACM-Rel1.0.0.exe /v"/l*v c:\temp\installer.log
µC/USB Device™ Class Vendor	ADI_uCUSBD_Class_Vendor-Rel1.0.0.exe /v"/l*v c:\temp\installer.log
µC/USB Device™ Class PHDC	ADI_uCUSBD_Class_PHDC-Rel1.0.0.exe /v"/l*v c:\temp\installer.log
µC/USB Device™ Class MSC	ADI_uCUSBD_Class_MSC-Rel1.0.0.exe /v"/l*v c:\temp\installer.log
µC/USB Device™ Class HID	ADI_uCUSBD_Class_HID-Rel1.0.0.exe /v"/l*v c:\temp\installer.log

License

The installation process checks for a valid license for each of the µC/USB Device™ Stack products as listed below. Refer to the Licensing Guide in your CCES installation which can also be found in http://www.analog.com/CrossCoreLicensingGuide.

µC/USB Device™ Stack for CrossCore®Embedded Studio Products
µC/USB Device™ Core for CrossCore®Embedded Studio Products
µC/USB Device™ Class CDC-ACM for CrossCore®Embedded Studio Products
µC/USB Device™ Class Vendor for CrossCore®Embedded Studio Products
µC/USB Device™ Class PHDC for CrossCore®Embedded Studio Products
µC/USB Device™ Class MSC for CrossCore®Embedded Studio Products
µC/USB Device™ Class HID for CrossCore®Embedded Studio Products

Support and Assistance

There are several options for contacting support:

- Submit your questions online at:

http://www.analog.com/support

- E-mail your Processor and DSP software and development tools questions from within CrossCore Embedded Studio.

Go to “Help->E-mail Support…”. This will create a new e-mail addressed to processor.tools.support@analog.com, and will automatically attach your CrossCore Embedded Studio version information (ProductInfo.html).

- E-mail your Processors and DSP applications and processor questions to:
- processor.support@analog.com OR
- processor.china@analog.com (Greater China support)
- Post your questions in the Processors and DSP online technical support community in Engineer Zone at

http://ez.analog.com/community/dsp

Supported processors

At the time of its release the only processor family supported in the released version of CrossCore Embedded Studio is ADSP-BF60x.

Software requirements

µC/USB Device™ Stack for CrossCore® Embedded Studio requires the support of an RTOS.

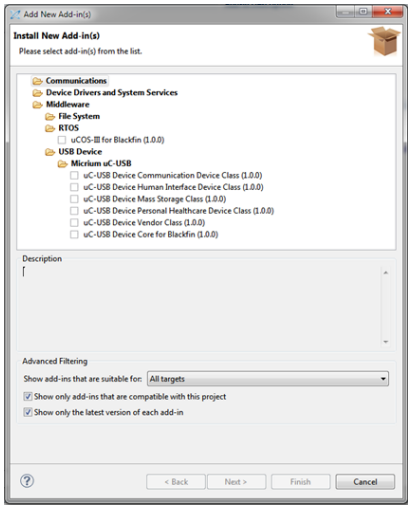
All of the examples ship with a pre-configured version of µC/OS-III™ Real Time Kernel for CrossCore® Embedded Studio. The RTOS is shipped in the form of a binary library that is linked into the examples. The exact configuration of the RTOS is provided in the example documentation.

6.2 Getting started with a project that uses µ C/USB Device Stack

Adding µC/USB Device Stack to a project

Every CrossCore Embedded Studio project contains a System Configuration file called system.svc which is located in the root of the project. The file is the IDE's interface for managing the various pre-written software components used in the "system" implemented by a project. Double-clicking any system.svc file in a navigation view opens that file in the

System Configuration Utility which allows you to see the add-ins that you currently have in your project. Clicking on Add and selecting one of the following add-in from the Middleware section under the USB Device category adds the selected product to your project.



Please note that **µC-USB Device Core for Blackfin (1.0.0)** is required for all other products (as previously discussed in this release note). Therefore, when you add in any one of the Class layer products, the µC/USB Device Core product will be automatically added in also.

If an RTOS has not been added in, when you select "Next" in the Add-In dialog, you will be presented with a warning screen indicating that an RTOS product does not yet exist in your application. You will not be able to proceed unless you also select **uCOS-III for Blackfin (1.0.0)** as an additional Add-In.

The µC/USB product add-ins generate code for initializing the µC/USB Device Stack. To ensure timely initialization, when system components are configured the IDE adds any required code to a global C function named adi_initComponents(). A call to this function will be added to the main()function when the USB and RTOS components are added.

Notes:

- Please refer to the µC/OS-III Release Notes for RTOS related information.

Configuration

µC/USB Device Stack application developers traditionally configure applications by creating header files which contain a long list of macro definitions. µC/USB Device Stack for CrossCore Embedded Studio provides a more intuitive configuration mechanism by providing a tab in the System Configuration utility, which can be accessed by double-clicking the system.svc file and selecting the µC/USB Device tab. Filling in all the required fields in the configuration tab generates the appropriate files, app_ucusbd_cfg.h and usbd_cfg.h located within the project under system/uc-USB/GeneratedSources.

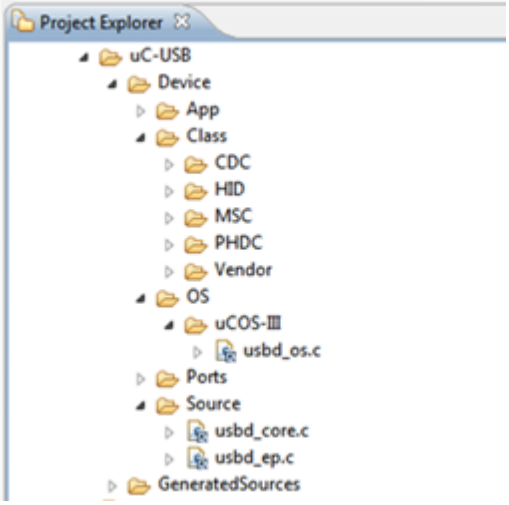
Notes:

1. Each Class Layer product will introduce a section within the µC/USB Device tab for Class Layer specific configuration. For example, the Mass Storage Class will introduce mass storage specific configuration options.
2. The defaults may not necessarily be appropriate for your application and you should set them to suit your needs. For example, the default setting for the maximum number of logical units supported, 2, may not be correct for your application. The configuration window tool tips should provide an indication of how to set the appropriate values, or look in the documentation for more details.

µC/USB Device Stack project structure

When adding µC/USB Device Stack to a CrossCore Embedded Studio project all the µC /USB Device Stack specific files get placed in the system folder. Please do not change this organization. In the system folder the following structure gets created

- A uC-USB folder. This folder contains sub-folders as follows



- A uC-CPU folder. This folder contains any sources and header files which are required by Micrium µC/CPU software. µC/CPU provides a processor-independent interface to the supported processors and toolchains that is used in all Micrium products.
- A uC-LIB folder. This folder contains any sources and header files which are required by Micrium µC/LIB software. µC/LIB provides a clean and organized implementation of some of the most common standard library functions, macros and constants. µC/LIB is required by many Micrium products including µC/USB.
- A uC-Common folder. This folder contains sources and headers which are common to several Micrium products but that are not part of any Micrium product themselves. These include app_cfg.h which is needed by all Micrium applications.

Examples

µC/USB Device Stack for CrossCore Embedded Studio provides examples which show how to use the various Device Classes. Each example is shipped for the ADSP-BF609 platform and can be used in both Release and Debug configurations.

Note:.

- Double-clicking on an example from the example browser or the system overview page opens the project in the installation folder without copying it to your workspace. If you want to modify the example in any way, it is recommended that it gets copied to your workspace. If you would like to copy the project to your workspace note that you may have to copy the sources separately. See Known Issues for more information.

Location

In order to locate µC/USB Device Stack examples and sketches, you can use the following:

- Open CrossCore Embedded Studio's (CCES) Example Browser which can be found in CrossCore Embedded Studio under Help. You may then perform one of the following steps
- In the Product Pull-Down select the USB Product that you have licensed and installed
- In the Keyword textbox insert the keyword "USB"
- The result of either of these filters will be a list of USB examples in the Search results panel. The results of browsing by USB keyword with all of the USB products installed is shown below

.

After locating an example of interest, double-clicking on the project in the search results pane will result in the example being imported into the CCES Project Explorer.

System view

CrossCore Embedded Studio provides the System Configuration Utility which is used by µC/USB Device Stack for CrossCore Embedded Studio. Use the System Configuration Overview tab to add the µC/USB Device Stack product add-ins, as appropriate, to a CrossCore Embedded Studio project.

To access the System Configuration Overview tab, do one of the following:

- In a navigation view, double-click the system.svc file of a project. The System Configuration utility appears with the overview tab selected.
- If the utility is already open, select the Overview tab.

As well as being able to add, remove and upgrade add-ins from this window, it also provides a list of examples and sketches associated with the selected add-in.

For more information about the System Configuration utility, see the CrossCore Embedded Studio help.

Configuration tabs

When the µC/USB Device Stack for CrossCore Embedded Studio gets added to a project, several configuration tabs become available in the System view. These include tabs for common Micrium components such as µC-LIB and µC-CPU. In addition, configuration tabs for µC/OS-III as outlined in the µC/OS-III release notes. These configuration tabs provide an easy mechanism to generate any macro definitions required by the Micrium products.

A µC/USB Device tab will also be added. Within the µC/USB Device tab a list of sub-tabs for each and every device class will be added.

For more information about each of the configuration options see the section µC/USB Device Tab in CrossCore Embedded Studio's help.

MISRA-C Support

MISRA C is a software development standard for the C programming language developed by the Motor Industry Software Reliability Association (MISRA). Its aims are to facilitate code safety, portability, and reliability in the context of embedded systems, specifically those systems programmed in ANSI C. The compiler detects violations of the MISRA rules at compile-time, link-time, and run-time.

As of this release a list of rules that µC/USB breaks is not available. The USB controller driver, provided by Analog Devices, suppresses the following MISRA rules

Rules 2.1, 8.5, 10.1.a, 10.1.b, 10.5, 11.5, 12.7, 13.7, 14.3, 14.7, 14.10, 16.2, 16.9, 16.10, 17.4, 18.4, 19.4, 19.10

µC/USB Device™ Stack for CrossCore® Embedded Studio RTOS Requirements

µC/USB Device Stack for CrossCore Embedded Studio requires the presence of an RTOS, although not necessarily µC/OS-III Real-Time Kernel for CrossCore Embedded Studio. When running in a µC/OS-III application, µC/USB Device Stack requires multiple µC/OS-III objects like semaphores and Task-specific registers slots.

Removing any of the µC/OS-III functionality that is required by a µC/USB Device application could cause link errors.

Note that adding µC/USB Device to a project which already has µC/OS-III may change some RTOS settings. The full list of specific changes is displayed by the µC/USB Device addition process.

Common Micrium Components

There are several CrossCore® Embedded Studio add-ins based on Micrium's products which share common components. To ensure that the same version of these components is used by all the add-ins that require them, these components are installed in a common location which is distinct from the add-in install folders. These common components are

- µC/CPU which is installed in %COMMONPROGRAMFILES%\Analog Devices\uC-CPU v1.0.0. This installation includes µC/CPU version 1.29.01.
- µC/LIB which is installed in %COMMONPROGRAMFILES%\Analog Devices\uC-LIB v1.0.0. This installation includes µC/LIB version 1.36.02.

The documentation for these components can be found in CrossCore® Embedded Studio Help under µC/OS-III™ 1.0.0 > Components Shared by Add-ins.

Known issues with µC/USB Device™ Stack for CrossCore® Embedded Studio

These are the currently known problems which affect µC/USB Device™ Stack for CrossCore® Embedded Studio.

TAR-48585: RTOS examples are not portable to other workspaces

The µCOS-III examples released with the µCOS-III product reference their source folder (src) and their readme in terms of a relative path from the project location. The reason for this was so the same sources and readme could be used for all processors that the example can be run on.
This method of linking files means that if a user chooses to import the project with the "Copy projects into workspace" box selected then the src and readme files are not copied and the example does not work.
If you want to import one of the examples to your workspace then you should follow these instructions:
Import the example copying it to your workspace. This step creates a new project in the workspace which contains two invalid links, to the readme and to the src folder.
Remove the existing link to the src folder and to the readme by selecting them on the new project and either press the "Delete" key or right-click and choose Delete.
Close the project.
Copy the readme and src folder to the location of the project in the new workspace. This will automatically add those files to the project. The location of the µCOS-III examples in the file system is: %RTOS_INSTALL_FOLDER%\uCOS-III\Examples.
Reopen the project
If the project was previously built in the installation folder then run "Clean" before rebuilding.
The example should build as work as expected.

NOTE: This issue also applies to the µC/USB Device Stack for CrossCore Embedded Studio products.

TAR-48536: µC/USB Device Stack for CrossCore Embedded studio can only be used with uCOS-III

µC-USB Device Stack for CrossCore Embedded studio is currently configured only to use µCOS-III for the RTOS. Micrium supply the OS layer files for µC/OS-II, which will be added in a future release of the product.