Release Notes for ADuCM4x50 EZ-KIT Board Support Package 1.1.0

Contents

Thank you for installing the ADuCM4x50 EZ-KIT® Board Support Package (BSP). The BSP provides software and documentation in support of the ADuCM4x50 EZ-KIT development board.

The BSP is designed to work with IAR Embedded Workbench software development tools.

For more details on IAR, please visit http://www.iar.com. The BSP provides comprehensive software support for the ADuCM4050 EZ-KIT development, including the drivers. The BSP also provides comprehensive examples which demonstrate the on-chip and off-chip device drivers. The device driver documentation is part of the BSP.

1 Required Software

To use this BSP in IAR Embedded Workbench environment, you must first obtain and install.

- IAR Embedded Workbench version 7.60.2 or later;
- Segger J-Link LITE v5.10p or later.

2 Release Testing

The BSP has been tested with

EZ-KIT	IAR
ADuCM4050 LFCSP EZ-KIT version 1.0 BOM Rev 1.2	J-Link Lite emulator
ADuCM4050 WLCSP EZ-KIT version 1.0 BOM Rev 2.1	J-Link Lite emulator

3 License Checking

Use of the BSP software is subject to the Software License Agreement presented during installation.

4 Installation

It is recommended that you backup or delete your older BSP installation directory

`(e.g. C:\Analog Devices\ADuCM4x50\ADuCM4x50_EZ_Kit)`

before installing a newer BSP version. The BSP installer does not currently offer an uninstall option.

5 Release Content

This release contains the following sets of components:

- Source files for device drivers. These components are authored by Analog Devices, for use on the ADuCM4x50 processor.
- Examples for device drivers. These components are authored by Analog Devices, and demonstrate the use of the device drivers. Some of the examples make use of the Micrium RTOS products or FreeRTOS products, so have a dependency on the Micrium products, which must be obtained separately under license from Micrium, or FreeRTOS.
- Toolchain support. These components are authored by Analog Devices, and are installed into the toolchain to configure it to recognize the ADuCM4x50 processor family.
- Templates to create ADuCM4x50 projects. When creating a new project, the release includes a no-OS and a ucos3 project template which add the appropriate macro definitions, include paths and sources to support the ADuCM4x50 processors.
- Cycle Counting Framework. A framework for measuring cycle counts has been added to the release. Documentation for cycle counting has been added to the "Modules" section. Specifically look for the "Cycle Counting Framework" section.
  As of this release only the SPI, CRC, SPORT and UART drivers have been instrumented to allow cycle counting. In future releases all of the drivers will be instrumented. Also, please note that the framework is only available in a non-RTOS context. It will be made available in a RTOS context in a future release.
- Additional utilities. These components are authored by Analog Devices, and assist in the generation of applications for the ADuCM4x50 processor family.
- Documentation.

5.1 Source files for drivers

ADuCM4x50.h	Device descriptions and macro files
System	Source and include files
Startup	Source and include files

Various peripheral device driver sources and include files in “Source” and “Include” directories.

5.2 Toolchain support files

The following common system infrastructure framework files are installed into the toolchain and should be used in all projects.

IAR tool chain configuration files

FlashADUCM4050.*	Flash loader files and sources
ioADuCM4x50.ddf	Debugger register display
ADuCM4050-flash.icf	ADuCM4x50 EZ-KIT linker control file for running the code from internal flash
ADuCM4050-sram.icf	ADuCM4x50 EZ-KIT linker control file for running the code from sram

5.3 Additional utilities

These utilities are installed into the tools subdirectory.

PinMuxUI	Java-based graphical utility for generating source code to configure pin multiplexing. Available for 32-bit and 64-bit Java Virtual Machines.
UartDivCalculator	Command-line utility for configuring the Baudrate for the UART device.

5.4 Documentation

ADuCM4x50_EZ-Kit_BSP_GettingStartedGuide.pdf	Getting Started Guide
ADuCM4x50_EZ-KIT_BSP_for_IAR_ReleaseNotes.pdf (this file)	Release Notes
ADuCM4x50_EZ-Kit_BSP_UsersGuide.pdf	User's Guide
ADuCM4x50_EZ-KIT_BSP_Device_Drivers_UsersGuide.pdf	Guidelines for using the Analog Devices device drivers.
ADuCM4050-EZ-KIT_Manual.pdf	ADuCM4050 EZ-Kit manual.
ADuCM4050-LFCSP-EZ-KIT-BOM.pdf	BOM for the ADuCM4050 LFCSP EZ-Kit.
ADuCM4050-LFCSP-EZ-KIT-Schematic.pdf	Schematics for the ADuCM4050 LFCSP EZ-Kit.
ADuCM4050-WLCSP-EZ-KIT-BOM.pdf	BOM for the ADuCM4050 WLCSP EZ-Kit.
ADuCM4050-WLCSP-EZ-KIT-Schematic.pdf	Schematics for the ADuCM4050 LFCSP EZ-Kit.
ADuCM4x50_ADuCM302x_Software_Differences.pdf	Specifies the differences in the API from the ADuCM302x drivers to ADuCM4x50 as well as a comparation of their footprint for a typical use case.
Html/index.html	Index file for HTML-based Device Driver API documentation

6 Examples

This section contains information about the example projects in the product.

6.1 Examples for drivers

Example projects have been provided for all on and off chip peripherals.

1.	ADC	• *adc_channel_read* : Demonstrate use of ADC Controller driver to sample the input signal at channel 0 and write the acquired samples to a file.
2.	ADXL363	• *accel* : Demonstrate how to use the ADXL363 driver to produce interrupts on activity events • *accel_using_FIFO* : Demonstrate how to use the ADXL363 driver, specifically using the on-chip FIFO to collect data
3.	Beeper	• *beeper_example*: Demonstrates the basic functionality of the beeper peripheral.
4.	CRC	• *core_driven_CRC* : Demonstrates how to use the CRC driver to compute the CRC, driven by the core. • *dma_driven_CRC_with_callback* : Demonstrates how to use the CRC driver to compute the CRC driven by the DMA with a callback function registered. • *dma_driven_CRC_without_callback* : Demonstrates how to use the CRC driver to compute the CRC driven by the DMA with no callback function registered.
5.	Crypto	• Demonstrates how to use Crypto device driver in various cipher modes.
6.	Cycle Counting	• *SPI_cycle_counting*: Demonstrates how to obtain cycle counts for the SPI driver.
7.	Flash	• *flash_block_protect* : Demonstrates the use of the Flash device driver flash memory block-protection feature. • *flash_page_write* : Demonstrates the use of the Flash device driver for flash memory data page write(s).

8.	FreeRTOS	• *SPI_Loopback* : demonstrates how to integrate the BSP with the FreeRTOS V9.0.0 as well as how to use the SPI driver in the context of the FreeRTOS.
9.	GPIO	• *LED_button_callback*: Demonstrates how to use the GPIO driver to Toggle LED's when the push buttons are pressed on the ADuCM4050 EZ-Kit.
10.	I2C	• *temperature_sensor*: Demonstrates how to use I2C driver for reading the data from the temperature sensor.
11.	Power_On_Self_Test	• Allows users to test the many peripherals of the EZ-Kit with push buttons to select specific tests to run.
12.	RNG	• *RNG_Example*: Demonstrates how to use and configure the RNG device for generating random numbers.
13.	RTC	• *Rtc_alarm*: Demonstrates how to use and configure the RTC device for generating the alarm periodically. • *Rtc_IO_example*: Demonstrates how to configure an RTC device to use the input capture and output compare features of RTC device.
14.	RTOS	• *ucos-III* : Demonstrates uCOS-III RTOS in ADuCM4x50 processors. This example demonstrates task and semaphore creation and context switching.
15.	SPI	• *LoopBack*: Demonstrates how to use the SPI driver in blocking mode and non-blocking mode with DMA and PIO. • *MasterSlaveLoopBack*: Demonstrates how to use SPI device for transmitting /receiving the data both in master and slave mode. • *RTOS_Callback*: demonstrates how to use the SPI device driver both in Master and Slave mode, with an RTOS and using Callbacks.
16.	SPI-flash	• w25q32_example: Demonstrate how to use the W25Q32 driver.
17.	SPORT	• *loopback_dma*: Demonstrates how to use the SPORT driver in DMA mode. • *loopback_int* : Demonstrates how to use the SPORT driver in PIO mode.
18.	SysTick	• *systick_example*: Demonstrates the use of SysTick timer to wait for a specific number of interrupts.

19.	TMR	• *tmr_example_gp*: Demonstrates how to use the General Purpose (GP) timers to generate a periodic interrupt and capture events. • *tmr_example_rgb*: Demonstrates how to use the Red-Green-Blue (RGB) timer to generate 3 PWM output signals with the same period but different duty cycles
20.	UART	• *Autobaud*: Demonstrates how to use UART device driver for baudrate detection. • *Callback*: Demonstrates how to to use UART in DMA mode, PIO mode and register a callback. • *Loopback*: Demonstrates how to use the UART driver to loop back the data between the TX and RX.
21.	WDT	• *wdt_example_interrupt*: Demonstrates how the Watchdog Timer (WDT) can be used to trigger an interrupt on timeout. • *wdt_example_reset:* Demonstrate how the Watchdog Timer (WDT) can be used both avoid and trigger a system reset.
22.	XINT	• *wakeup_button*: Demonstrates the use of XINT driver to Toggle LED when the wakeup button is pressed on the ADuCM4x50 EZ-Kit.

6.2 Example for Micrium Components

The example "ucos-III" in the examples\rtos directory relies on the uC/OS-III RTOS component from Micrium - refer to the example's README file for details.

To make use of this example:

1. Obtain the uC/OS-III product from Micrium. the integrated cortex-m port released with version 3.06.00 is recommended. This version requires the following minimum versions:
   - uC-CPU V1.31.00
   - uC-LIB V1.38.02
2. Choose a directory where you will store the unzipped Micrium source files.
3. Set the Windows environment variable ADUCM4x50_MICRIUM_DIR to the pathname of this directory.
4. Unzip all Micrium components into `$ADUCM4x50_MICRIUM_DIR`
5. Start the IAR Embedded Workbench environment.

6. Import the appropriate example, and build it. The example is configured to make use of the `$ADUCM4x50_MICRIUM_DIR` environment variable.

Please note that the Micrium components are licensed products, and you must obtain the necessary licenses directly from Micrium to use them.

6.3 Example for FreeRTOS Components

The example "SPI_Loopback" in the examples\FreeRTOS directory relies on FreeRTOS version 9.0.0 - refer to the example's README file for details.

To make use of this example:

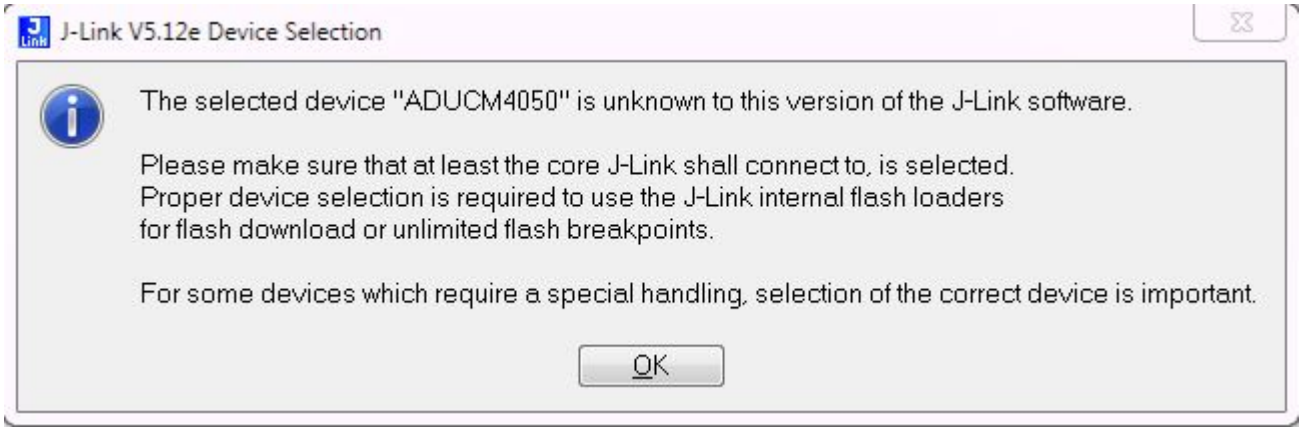
1. Obtain FreeRTOS version 9.0.0 from http://www.freertos.org.
2. Follow the step describe in the Readme_freertos_spi_loopback.txt file for IAR.

7 Location

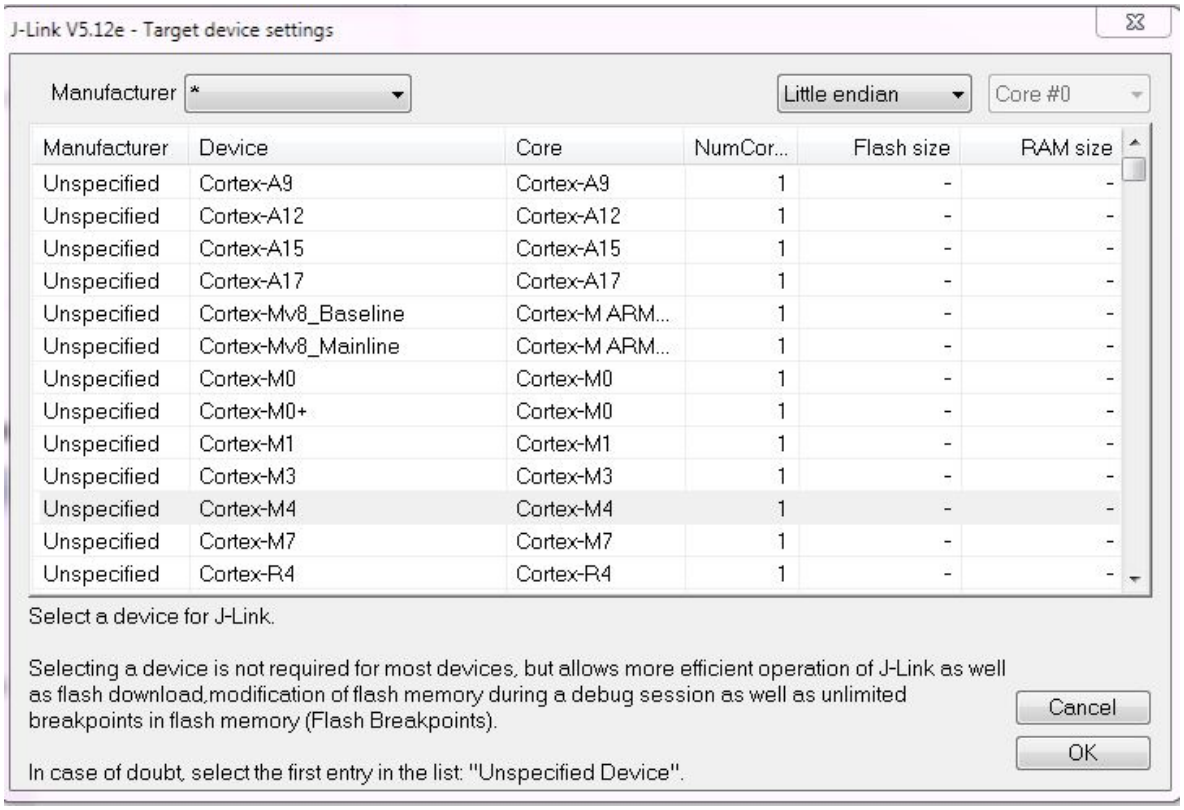
By default, the BSP will be installed into the directory `C:\Analog Devices\ADuCM4x50\ADuCM4x50_EZ_Kit`.

8 J-Link Device Selection

IAR Embedded Workbench 7.60.2 ships with J-Link Software 5.12e. This version of J-Link Software does not recognize the ADuCM4050, and the following pop-up windows will appear the first time an executable is loaded to the target in IAR. Press "OK" when the following window appears:



Choose "Unspecified Cortex-M4" when the subsequent window appears:



Alternatively, these windows can be avoided by updating the version of J-Link Software on the host machine to version 6.10a or later, where the ADuCM4050 is a recognized device. The update can be done by visiting the SEGGER website and downloading the J-Link Software installer. The installer will update all existing J-Link files on the host machine, after prompting the user with a list of the existing IAR install directories. Make sure to check the IAR install directory containing version 7.60.2 in order for these files to be updated.

9 Device Driver Thread Safety

All Device Drivers are **not** thread-safe. They are re-entrant but not thread-safe. If an RTOS is present, then drivers will use the RTOS semaphores for implementing the blocking calls.

10 Contacting Technical Support

You can reach Analog Devices software and tools technical support in the following ways:

- Post your questions in the software and development tools support community at EngineerZone®.
- E-mail your questions about processors and processor applications to processor.support@analog.com.
- For Greater China, Processors and DSP applications and processor questions can be sent to: processor.china@analog.com.
- Submit your questions to technical support directly via http://www.analog.com/support.
- Contact your Analog Devices sales office or authorized distributor.
- For IAR tool chain support please visit: http://www.iar.com/support

11 Known Issues

For the latest anomalies please consult our Software and Tools Anomalies Search page.

11.1 RTC Example RTC_Alarm (MSKUV01-99)

When executing this example, it should be let to run till completion, until "All done!" is seen. If the execution is stopped midway, then it will lock up the board.

If this happens, the board can be unlocked with the following actions: Hold the Boot Button(SW1) and Toggle the Reset Button(SW2) twice and then release the Boot Button(SW1).