



ADuCM350BBCZ Device Drivers Library

Release Notes

Release 2.4.0.0 - December, 2015

Product Information	
Name	ADuCM350BBCZ
Release Number	2.4.0.0
Release Date	December, 2015
Support	http://ez.analog.com/community/analog-microcontrollers/aducm350

NOTE: The RTF version of this file is best viewed with Microsoft® Word or other word processor that supports modern style and formatting content better than Microsoft® WordPad... better still; view the PDF version located in the same directory as the RTF version.

ADuCM350BBCZ Device Drivers Library Release Notes

RELEASE HISTORY		
Release Number	Date	Purpose
2.4.0.0	December, 2015	<p>Please see the 2.4.0.0 release section for detailed information. Highlights of the release includes fixes/features for the following:</p> <ul style="list-style-type: none">• SPI hardware chip select APIs added• ThreadSPIMasterSlave example added• SPI Device Driver issues addressed
2.3.0.0	December, 2014	<p>Please see the 2.3.0.0 release section for detailed information. Highlights of the release includes fixes/features for the following:</p> <ul style="list-style-type: none">• USB/RTOS related modifications• SPI Device Driver• Examples removed:
2.2.0.0	March, 2014	Initial GP release of Software Device Drivers Library for ADuCM350BBCZ.

ADuCM350BBCZ Device Drivers Library Release Notes

INTRODUCTION 2.4.0.0 RELEASE

This document contains the *Release Notes* for the ADuCM350BBCZ Device Drivers Library, version 2.4.0.0. It subsumes the previous release notes which are included in this document.

This release will support the IAR support files for IAR version 7.30.1 and above.

HIGHLIGHTS

New features

- The SPI driver now supports both Hardware and Software chip select
- A new example, interfacing to the FTDI FT422H SPI-to-USB controller, has been added
- A new example, ThreadSPIMasterSlave, has been added. This example demonstrates how to use the SPI APIs in both Master and Slave contexts. It works in both PIO and DMA modes. It is an RTOS based example.

Issues Addressed

- AFE driver
 - AFE interrupt handlers are now passing the clear and/or disable bits to the associated callback.
 - The AFE_Examples.eww file is now included in the kit.
- The GPT API adi_GPT_GetTimerBusy() return value is now documented correctly.
- AFE Interrupt Handlers now pass clear/disable bit to the callback.
- Doxygen documentation now includes all APIs
- GPIO driver issues
 - The GPIO driver adi_GPIO_RegisterCallback() API documentation has been corrected. The GPIO callback does not need to clear pin interrupts before returning as the CommonInterrupt Handler does this.
 - The GPIO driver APIs, adi_GPIO_SetGroupInterruptPolarity and adi_GPIO_SetGroupInterruptsPins , now allow Pins==0 when ADI_DEBUG is defined.
- A file, metrics.h, was missing from the BSP and has been added.
- Various SPI driver issues were fixed
 - In PIO mode the SPI driver was sending out an additional byte for each transceiver call
 - In PIO mode blocking calls were incorrectly polling in RTOS mode. The code was changed such that an RTOS semaphore pend call is made.
 - An issue switching from DMA to PIO mode resulting in a transaction failure has been addressed. The SPI driver now works correctly when DMA mode is dynamically turned on or off.
- Missing I2C API Doxygen generated documentation is now included .
- UART driver issues

ADuCM350BBCZ Device Drivers Library Release Notes

- The UART driver and system.c sources have been changed such that it will compile without error with gcc (GNU builds are not supported by ADI, but the sources can easily be setup to build with gcc).
- The UART driver has been modified such that the receive buffer interrupt is not enabled by default.
- The UART DivMap now has a definition for 20 MHz.
- The LCD driver data structure was inconsistently declared. The declarations are now consistent.
- Erroneous references to a file that does not exist (adi_pwr_xxx APIs in pwr.c) have been purged from system.c, gpt.c and wut.c.
- CRC buffers of less than 1024 bytes are now being processed correctly.

KNOWN ISSUES

- When using the SPI driver in DMA mode the increment parameters are ignored. If this functionality is needed then PIO mode must be used.
- When using the Captouch_5ButtonsTest example with Eval-ADUCM350EBZ, it is necessary to remove 0 Ohm resistors R63 and R65 in order to get full functionality.
- The SPI1 SCLK pin can also be used as an external GPIO. For this reason, a push-button switch (S4) has been implemented on the Eval-ADUCM350EBZ motherboard. This button has an associated de-bounce capacitor (C5) and this cap is can affect the SPI1 SCLK. The solution is to remove 0r resistor R12 from Eval-ADUCM350EBZ, which will isolate the cap and switch.

ADuCM350BBCZ Device Drivers Library Release Notes

INTRODUCTION 2.4.0.0 RELEASE

This document contains the *Release Notes* for the ADuCM350BBCZ Device Drivers Library, version 2.4.0.0. It documents the requirements, install procedure, bug fixes and unresolved issues, as well as any late-breaking information to supplement the main documentation. The companion document “*ADuCM350BBCZ Device Drivers Getting Started Guide*” provides more in-depth information concerning the device drivers and usage. The complete *Device Driver API* is fully documented (in html format) within the release itself.

HIGHLIGHTS

This release provides a comprehensive ADuCM350BBCZ Software Device Driver Library package. The ADuCM350BBCZ integrates an ARM Cortex-M3 microcontroller with various on-board peripherals and Metering-on-a-Chip (MoC) technology within a single package. The following device drivers and application examples are included in this release.

Feature Added and Issues Addressed

- USB/RTOS
 - Added support for the USB Communications Device Class (CDC) with accompanying example.
 - Updated support for Micrium’s latest uC/OS-II release (4.04.01).
 - Updated to the latest USB driver from ADI, featuring the following improvements:
 - USB host and device modes are integrated
 - DMA mode 1 is fully functional
 - Improved throughput
 - Removed obsolete bsp directory and files and revised examples to build without them.
- Removed deprecated dynamic GPIO pin multiplexing APIs from the GPIO service. A static pin multiplexing GUI is now used to reduce target code footprint and code complexity. (See “Static Pin Multiplexing” in “ADuCM350BBCZ Software Users Guide” for details.)
- Removed “TEST_COMMON_USES_UART” STDIO redirection macro. Any UART output must be managed at the application level -- after first configuring the UART pins as required as part of static pin mux configuration process (See “PinMuxUI” in the “tools” directory of this release). UART output has been added at application level for Analog Front End (AFE) based examples to facilitate returning results from the ADC.

ADuCM350BBCZ Device Drivers Library Release Notes

- SPI Device Driver
 - Added support for designating any GPIO pin for use as SPI chip select.
 - Added DMA data chunking to schedule large data transactions into a chain of smaller DMA descriptor blocks, as needed. The DMA “ping-pong” cycle type is used to automatically switch/update descriptors. This is entirely within the SPI driver and transparent to the user.
 - Enforced DMA mode size checking of the “DataSize” member in the ADI_SPI_TRANSCEIVE_TYPE structure (used by both adi_SPI_MasterTransfer() and adi_SPI_SlaveTransfer() APIs) to insure evenly-size counts and buffer allocations for DMA transactions. The 8-bit byte count must be evenly-sized because underlying SPI controller employs only 16-bit DMA mode.
 - Added DMA error handlers to the SPI examples to intercept “invalid” DMA descriptor interrupts resulting from the *intentional* use of invalid DMA cycle type to mark the end of a multiple DMA descriptor chain. Such errors are allowed by the handler while other DMA errors are trapped (bus errors, etc.)
- Examples removed:
 - SpiMasterSlaveTest example removed. If this example is required, please contact ADI. See Support section.
 - Mipi_SSD1963_Test example removed due to hardware supply issue.

ADuCM350BBCZ Device Drivers Library Release Notes

ADUCM350BBCZ DEVICE DRIVERS

This release includes the following device drivers and modules:

- AFE - Analog Front End
- BEEP - Beeper
- CAPTOUCH - Capacitive Touch Sensor
- CRC - Cyclic Redundancy Check controller
- DMA - Direct Memory Access
- FLASH - Flash Memory Controller
- GPIO - General Purpose Input/Output Controller
- GPT - General Purpose Timers
- I2C - Inter-Integrated Circuit Serial Interface
- I2S - Integrated Inter-Chip Sound Digital Audio Interface
- LCD - Liquid Crystal Display Controller
- OSAL - Operating System Abstraction Layer
- PDI - Parallel Display Interface
- MIPI - Mobil Industry Processor Interface
- RNG - Random Number Generator
- RTC - Real-Time Clock Interface
- SPI - Serial Peripheral Interface
- Startup - IVT initialization/relocation, default interrupt handlers, reset handler
- System - System initialization, clock, clock mux, clock divide, power and hibernation control
- UART - Universal Asynchronous Receiver/Transmitter
- USB - Universal Serial Bus
- WDT - Watchdog Timer
- WUT - Wakeup Timer

EXAMPLES

This release includes numerous application examples illustrating the use of the device drivers with the ADuCM350BBCZ. Each test has a companion Readme.txt file, documenting test setup and operation. The examples also contain IAR build directories with required project files, workspaces, etc., needed to build each example.

SEMI-HOSTED EXAMPLE OUTPUT

In general, example output is printed to STDIO, which is reported to the debugger console via the semi-hosting facility (see the "Capturing STDIO from Examples" section of the "ADuCM350BBCZ Software Users Guide").

ADuCM350BBCZ Device Drivers Library Release Notes

Typically, each such example prints "PASS!" or "FAIL: <filename, linenumber, message>" messages, depending on success or failure of the example run. Some tests provide additional performance data or message reports during the test run. Performance messages are formatted as "PERF: <message>".

Semi-hosting is enabled by default in most examples.

UART EXAMPLE OUTPUT

Special-case examples (e.g., Analog Front End (AFE) based examples and the Uart_CharEcho example) report data/results via the Segger UART-to-USB transceiver on the USB-SWD/UART-EMUZ emulator board. See the respective readme.txt files for additional instructions.

For further information on the UART-to-USB hardware and software, see the “ADuCM350BBCZ Devices Drivers Getting Started Guide”. To capture these results, please follow the instructions in the “UART-to-USB Transceiver Software Installation” section of the Getting Started Guide, making sure to set up and run a PC based terminal application session (or other communications port monitor) to view the output.

The Segger USB Serial Converter requires a PC device driver installation prior to operation.

ADuCM350BBCZ Device Drivers Library Release Notes

REQUIREMENTS

SYSTEM REQUIREMENTS

The minimum system requirements for running the examples and applications are as follows:

PC with USB 2.0 controller connection and one free (virtual) COM Port.

Windows 7 Enterprise SP1 or later.

SOFTWARE REQUIREMENTS

The released software requires the “IAR Embedded Workbench®” (**revision 7.30.1**) installed to build projects. The ADuCM350BBCZ Device Driver Software installer adds various system files to the IAR installation, enabling it to support the ADuCM350BBCZ. The Keil toolchain is not supported.

Some of the software (RTOS, USB, etc.) depends on a set of Micrium-provided components described in the “Release Content” section of this document. The Micrium-provided components are to be obtained and installed in a separate common directory, pointed to by the Windows User Environment Variable: “**MICRIUM_DIR**”.

HARDWARE REQUIREMENTS

Analog Devices’ “USB-SWD/UART-EMUZ” Emulator Board, “mIDAS-Link” or IAR “jLink” emulator pod for debugging. Segger “J-Link Lite” and IAR “jTrace” emulators are also supported. These are all either Segger or Segger-derived emulators and will self-install Windows drivers on initial connection. They will all also self-prompt to update firmware if they find newer firmware updates available. Generally, we recommend applying the firmware updates if prompted.

Evaluation board(s) - See below for approved hardware revisions:

Supported Hardware	Board Rev
EVAL-ADUCM350EBZ	B

For detailed information on using the EVAL-ADUCM350EBZ board, including jumper configuration and information, please see the EVAL-ADUCM350EBZ User Guide (UG-668)¹.

ADuCM350BBCZ Device Drivers Library Release Notes

The GPIO pins on the ADuCM350 reside on different power supply domains, depending on what their secondary function is. The I/O supply domains for the different pins on the ADuCM350 are documented in the Pins section of the Hardware Reference Manual (UG-587, table 635 in Rev.A)¹.

On the EVAL-ADUCM350EBZ motherboard, the M5 and LK14 jumpers can be used to connect the VDD_IO and VDD_LCD domains respectively.

M5 JUMPER: VDD-IO SOURCE SELECT

If M5 is in the “A” position, VDD_IO will be supplied from VCCM_ANA. If M5 is in the “B” position, VDD_IO will be supplied from the 3.3V regulator on the motherboard.

LK14 JUMPER: VDD-LCD SOURCE SELECT

If LK14 is shorted, VDD_LCD will be supplied from VCCM_ANA. LK14 must be open when using the LCD segment driver. See the LCD Controller section of the Hardware Reference Manual for more information.

¹ See the References section of the ADuCM350BBCZ Software Users Guide.

ADuCM350BBCZ Device Drivers Library Release Notes

INSTALL PROCEDURE

Follow the instructions in the IAR *Embedded Workbench for ARM* product install manuals for installation of the supported tool chain software.

After installation of the IAR IDE, proceed to install ADI ADuCM350BBCZ Device Drivers using the provided Windows installer program.

The ADuCM350BBCZ device driver installer prompts to select the supported host tool chain; please choose something compatible with the Software Requirements section. The preferred device driver install location is also prompted, with the default choice being *outside* (see note below) the tool chain directory. Users may install the device drivers anywhere; a tool chain is not required to install a “driver-only” file set. Only tool chains meeting the minimum release versions specified in this release note are offered.

NOTE: Windows 7 and Windows Vista: If the device drivers and example/test projects are installed in the (now protected) “Program Files” directory area, i.e., within the tool chain directory, both the IAR and Keil Integrated Development Environments (IDE) must be “**Run as Administrator**”. This is because “Program Files” is a protected Windows directory. Running either toolchain “as administrator” is not necessary if the device drivers are installed in a directory outside the “Program Files” tree. *ADI recommends **not** installing the drivers under the “Program Files” directory area in order to avoid access problems.* This is the default behavior of the installer.

ADuCM350BBCZ Device Drivers Library Release Notes

RELEASE CONTENT

There are two domains of files comprising this release, which work together: **ADI-generated** files (i.e., this release); and **Micrium** files, described below under the “Micrium Base Files” section. The Micrium files are to be obtained directly from Micrium. The ADI-generated files are summarized in the target doc directory in file: manifest.xml. The manifest file controls the content of the ADI release and should accurately reflect all ADI content placed by the installer. *ADI does not deliver the Micrium files*, which are licensed commercial products to be obtained directly from Micrium.

ADI-GENERATED FILES

The following common system infrastructure framework files are installed and should be used in all projects. All files (IAR and driver) are installed in the toolchain directory tree by default. Driver files can be installed anywhere.

IAR tool chain configuration files:

ioADUCM350BBCZ.ddf	debugger register display
ADUCM350BBCZ.i79	device database
ADUCM350BBCZ.menu	device database
ioaducm350BBCZ.h	debugger register modify directives
ADuCM350BBCZ.icf	Generic ADuCM350BBCZ linker control file
FlashADUCM350BBCZ.*	Flash loader files and sources

Device Drivers, system, startup, IVT mapping, includes, etc., files:

adi_int/adi_nvic	source and include files
adi_types.h	base typedefs
ADuCM350.h	device descriptions and macro files
System	source and include files
Startup	source and include files
device.h	ADuCM350 memory mappings
macros.h	register bitfields, masks, etc.
Plus various peripheral device driver source and include files in “src” and “inc” directories.	

OSAL[®] files:

uCOS-II	ADI-provided Operating System Abstraction Level for uCOS-II
noos	ADI-provided OSAL stub for non-OS systems

USB[®] files:

USBD	ADI USB device files
------	----------------------

Tools:

IEIfTool	Modified IAR elf utility with new support for parity and reverse CRC signatures
PinMuxUI	Graphical pin mux configuration and code generation utility
SerialDownloader	Serial port flash downloader utility

Doc:

ADuCM350BBCZ Device Drivers Getting Started Guide (pdf)

ADuCM350BBCZ Device Drivers Library Release Notes

ADuCM350BBCZ *Device Drivers Release Notes* (pdf, rtf) (this file)

ADuCM350BBCZ *Software Users Guild* (pdf)

Html/index.html

Index file for HTML-based Device Driver API documentation

*Some of these files are based on examples provided by Micrium which are intended to serve as an example template to be modified as-needed. In all cases, such files are identified with an “EXAMPLE CODE” comment from Micrium in the file header, declaring them as such.

MICRIUM COMPONENTS

These components are intended to come directly from Micrium and to be used “out of the box”, as-is. *ADI does not deliver the Micrium files.* The Micrium content is entirely disjoint from the rest of the ADI release for the ADuCM350. To get the examples to build against the Micrium components, unzip all the Micrium zip files into the *same*, common directory and set a Windows system **environment variable** named “**MICRIUM_DIR**” to point to the Micrium “Software” directory node (the *singular, common destination directory where the zip files were expanded*). This environment variable is used by the ADI-provided IAR example project files to resolve the Micrium source location. Details of the required Micrium components are:

Module	Version	Release Date	File Size	File Name
uC/OS-II Core	2.92.11	2014-04-07	17584142	KRN-K2XX-PKG000-X-P1.zip
uC/USB-Device Core	4.04.01	2014-09-03	364.94 K	USB-USBD-000000-X-P1.zip
uC/USB-Device CDC	4.04.01	2014-09-03	29.18 K	USB-USBD-CDCACX-X-P1.zip
uC/USB-Device MSC	4.04.01	2014-09-03	55.33 K	USB-USBD-MSCX-X-P1.zip
uC/USB-Device PHDC	4.04.01	2014-09-03	8.72 M	USB-USBD-PHDCXX-X-P1.zip

All of these files are licensed products from Micrium and licenses must be obtained from Micrium to use them. See the Build Configurations chapter of the ADuCM350BBCZ Software User’s Guide for more information on using Micrium components.

OPERATING SYSTEM ABSTRACTION LAYER (OSAL)

The device drivers and system services provide a Real-Time Operating System (RTOS) abstraction layer (OSAL) that supports operation with or without an RTOS. Micrium’s uC/OS-II and a non-RTOS mode are currently supported. Each of the typical RTOS objects/operations is mapped through the OSAL so that different RTOS implementations may be accommodated with minimal impact on application code. To use the OSAL, an application must call the “adi_osal_Init ()”.

Please reference the ADuCM350BBCZ Software User’s Guide for further details on using the OSAL.

ADuCM350BBCZ Device Drivers Library Release Notes

USB-IF COMPLIANCE TESTING

The USB implementations provided in this release have been tested under the USB-IF (Implementers Forum) Compliance Program, (see <http://www.usb.org/developers/compliance>). Formal compliance certification is not claimed.

All of the Chapter 9 Tests (revision 7050) are passing.

All of the MSC Tests (revision 7073) are passing except for the Command Set Tests, which are failing because this test performs read and write operations up to 64k bytes where the provided MSC example can only provision for 2.5k bytes because of RAM availability on the target; a sufficiently large RAM Disk cannot be allocated to meet the needs of the test.

All of The PHDC Tests (revision 7073) are passing.

DEVICE DRIVERS MEMORY FOOTPRINT DATA

Representative ADuCM350BBCZ Device Drivers Library memory footprint data is provided here as an aid in estimating application memory requirements. Actual results will vary depending on how much of each driver API is used.

The footprint data intentionally omit application space, system startup, and system run-time overhead, which will vary by application demands, compiler optimizations, linker elimination, stack and heap allocations, etc.

The following assumptions were made in collecting this data:

- The footprint data are derived as an average of various example application projects included in this release.
- The data are collected from release-mode builds with full optimization (preference set for size over speed) and linker elimination enabled.
- The example programs do not necessarily reference each and every driver or driver API.
- The data only assesses device driver footprint; no assessment of system startup code, heap or stack allocations, interrupt vector table size, default interrupt handlers, etc., impact is made.

ADuCM350BBCZ Device Drivers Library Release Notes

The table data is taken over an average of all the examples; though all drivers and the entire API of any one driver may not be used by any one user application. Application requirements are excluded in this summary and must be considered separately as part of a complete memory requirements analysis.

Representative ADuCM350BBCZ Device Drivers Library Memory Footprint Data

(Average values of Examples in release build mode, as of release 2.4.0.0)

Device Driver Object File	Flash Code (ROM)	Flash Data (ROM)	RAM Data
afe.o	2057	0	159
afe_lib.o	735	0	0
beep.o	360	0	20
captouch.o	1398	37	20
captouch_lib.o	432	0	0
crc.o	708	0	48
dma.o	492	13	530
flash.o	569	71	151
gpio.o	343	3	92
gpt.o	303	0	48
i2c.o	1549	4	150
i2s.o	888	6	134
lcd.o	341	14	20
pdi.o	224	0	8
rng.o	308	0	20
rtc.o	482	0	20
spi.o	1682	6	315

ADuCM350BBCZ Device Drivers Library Release Notes

uart.o	343	43	47
wdt.o	292	1	4
wut.o	401	0	20

Note, USB footprint data has been omitted, as it is highly Micrium and application-dependent. Application specific footprint information for each of the USB examples (uCOS-II-USBD-CDC, uCOS-II-USBD-MSC and uCOS-II-USBD-PHDC) has been provided in the readme associated with each example.

KNOWN ISSUES

This is a full product release. Please consult the individual “Readme.txt” files in each example folder for operational details and expected behavior of each example.

The following are known issues with the current release:

- “User” environment variables within IAR project files: Non-IAR user macros within IAR project files -- e.g., “\$MICRIUM_DIR\$” (translating to the “MICRIUM_DIR” system environment variable) in various “*.ewp” example project files provided with this release -- are not preserved when the project file settings are saved by IAR. IAR appears to preserve their own such macros (e.g., “\$PROJ_DIR\$”, “\$TOOLKIT_DIR\$”, etc.), but they do not preserve user-defined macros with the current IAR release; user-defined relative-path macros are expanded to fully-qualified hard paths on the project file rewrite. Such modified project files become non-portable due to hard-coded paths.
- Occasionally, the IAR IDE and/or Segger emulator will present a connectivity problem with the target. It is not known exactly what the actual problem is, though one scenario is always sure to cause problems: resetting the target during live debug sessions. Generally, the emulator (J-Link) will pop-up an error message saying “Communication timed out: Requested 4 bytes, received 0 bytes! Abort debug session? (yes/no)”. If and when this happens, it is usually recovered by simply recycling the power to the emulator by pulling the USB connections and reinserting it.
- Occasionally, when the ADuCM350 is configured as a USB MSC device (using the uCOS-II-USBD-MSC example) and is connected to a host PC, it may take up to 1 minute for the host PC to detect the MSC device.
- When executing the PHDC application example (uCOS-II-USBD-PHDC) in multiple mode, upon pressing 2 to exit, the user may see ERROR #995. In this scenario, reads may have been issued before exit, but on exit these reads fail and generate an error message.

ADuCM350BBCZ Device Drivers Library Release Notes

TOOL CHAIN AND HOSTING ISSUES

TOOLCHAIN INSTALL SEQUENCE

If the ADuCM350BBCZ Device Driver Software is installed without a code generation toolchain present, the Device Drivers will be installed in a user-defined location, without the required toolchain-specific system files.

If a build toolchain install is planned *after* the ADuCM350BBCZ Device Driver Software has been installed, it is necessary to reinstall the ADuCM350BBCZ Device Driver Software (after the toolchain install) so the required ADuCM350BBCZ product activation files may be populated in the toolchain directories.

ADuCM350BBCZ Device Drivers Library Release Notes

TECHNICAL CONTACTS AND SUPPORT

You can reach Analog Devices, Inc. Customer Support at:

- Visit the EngineerZone ADuCM350 Design Support Community at <http://ez.analog.com/community/analog-microcontrollers/aducm350> for FAQs and a forum for discussion with Analog Devices Applications Engineers.
- Visit the ADuCM350 Product Page at <http://www.analog.com/en/processors-dsp/analog-microcontrollers/aducm350/products/product.html> to order hardware and for software and documentation downloads.
- For IAR tool chain support please visit <http://www.iar.com/support>
- Phone questions to 1-800-ANALOGD
- Contact your Analog Devices, Inc. local sales office or authorized distributor
- Send questions by mail to:
Analog Devices, Inc.
3 Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
USA