



Release Notes for ADuCM4x50 Device Family Pack 3.2.0

Contents

1	Introduction	3
2	Release Notes for ADuCM4x50 Device Family Pack 3.2.0	4
2.1	Differences between version 3.2.0 and prior versions	4
2.1.1	ADC	4
2.1.2	Crypto	4
2.1.3	RTC	4
2.1.4	SPI	6
2.1.5	TMR	6
2.1.6	UART	6
2.1.7	Vector Table	7
3	Release Notes for ADuCM4x50 Device Family Pack 3.1.2	8
3.1	Differences between version 3.1.2 and prior versions	8
3.1.1	Silicon Revision	8
3.1.2	GPIO	8
3.1.3	PWR Driver	8
3.1.4	RTC Driver	8
3.1.5	RTOS	9
3.1.6	UART Driver	9
4	Release Notes for ADuCM4x50 Device Family Pack 3.1.0	11
4.1	Differences between version 3.1.0 and prior versions	11
4.2	Required Software	11
4.2.1	Keil uVision	11
4.2.2	CrossCore Embedded Studio	11
4.2.3	IAR Embedded Workbench	11
4.3	Release Testing	12
4.3.1	Keil uVision	12
4.3.2	CrossCore Embedded Studio	12
4.3.3	IAR Embedded Workbench	12
4.3.4	License Checking	13
4.4	Release Content	13
4.4.1	Source files for device family drivers	13
4.4.2	Location	13
4.4.3	Device Driver Thread Safety	14
4.4.4	Contacting Technical Support	14
4.4.5	Examples	15
4.5	Known Issues	15

1 Introduction

Thank you for installing the ADuCM4x50 Device Family Pack (DFP). This document describes the changes for the ADuCM4x50 Device Family Pack 3.2.0 ADuCM4x50 Device Family Pack 3.2.0 is supported in Keil uVision, CrossCore Embedded Studio® (CCES) and IAR Embedded Workbench.

2 Release Notes for ADuCM4x50 Device Family Pack 3.2.0

2.1 Differences between version 3.2.0 and prior versions

ADuCM4x50 Device Family Pack has been updated with software modifications MSKCC-69, MSKUV-210, MSKUV-214, MSKUV-257, MSKUV-261, MSKUV-262, MSKUV-263, MSKUV-264, MSKUV-265, MSKUV-266, MSKUV-267, MSKUV-270, MSKUV-294, MSKUV-305, MSKUV-318, MSKUV-322 , MSKUV-328, MSKUV-329, MSKUV-331, MSKUV-334, MSKUV-335, MSKUV-336.

2.1.1 ADC

MSKUV-336

Documentation for SetAcquisitionTime function improved.

2.1.2 Crypto

MSKUV-328

The crypto driver has been made common to ADuCM302x and ADuCM4x50 families.

MSKUV-331

Allow 0-length input data for CCM mode.

2.1.3 RTC

MSKCC-69

Adding pend before write and sync after write for RTC registers that require it in rtc_init, and remove obsolete comments.

MSKUV-210

Proper support for function adi_rtc_SetAutoReloadValue on ADuCM4x50 implemented.

MSKUV-214

Support to dynamically configure the input capture overwrite enable feature.

MSKUV-264

Function `adi_rtc_SetAlarmRegs` added.

MSKUV-265

Function `adi_rtc_SetSensorStrobeChannelMask` now preserves previous settings.

MSKUV-266

Proper RTC interrupts support for both ADuCM302x and ADuCM4x50.

MSKUV-267

Missing RTC static configuration added: `RTC1_CFG_CR5SSS_OC1SMPEN`, `RTC1_CFG_CR5SSS_OC1SMPMTCHIRQEN`, `RTC1_CFG_CR5SSS_OC2SMPEN`, `RTC1_CFG_CR5SSS_OC2SMPMTCHIRQEN`, `RTC1_CFG_CR5SSS_OC3SMPEN`, `RTC1_CFG_CR5SSS_OC3SMPMTCHIRQEN`, `RTC1_CFG_CR6SSS_OC1SMPONFE`, `RTC1_CFG_CR6SSS_OC1SMPONRE`, `RTC1_CFG_CR6SSS_OC2SMPONFE`, `RTC1_CFG_CR6SSS_OC2SMPONRE`, `RTC1_CFG_CR6SSS_OC3SMPONFE`, `RTC1_CFG_CR6SSS_OC3SMPONRE`, `RTC1_CFG_CR7SSS_OC1SMPEXP`, `RTC1_CFG_CR7SSS_OC1SMPPTRN`, `RTC1_CFG_CR7SSS_OC2SMPEXP`, `RTC1_CFG_CR7SSS_OC2SMPPTRN`, `RTC1_CFG_CR7SSS_OC3SMPEXP`, `RTC1_CFG_CR7SSS_OC3SMPPTRN`, `RTC1_CFG_GPMUX0_OC1GPIN0SEL`, `RTC1_CFG_GPMUX0_OC1GPIN1SEL`, `RTC1_CFG_GPMUX0_OC1GPIN2SEL`, `RTC1_CFG_GPMUX0_OC2GPIN0SEL`, `RTC1_CFG_GPMUX0_OC2GPIN1SEL`, `RTC1_CFG_GPMUX1_OC2GPIN2SEL`, `RTC1_CFG_GPMUX1_OC3GPIN0SEL`, `RTC1_CFG_GPMUX1_OC3GPIN1SEL`, `RTC1_CFG_GPMUX1_OC3GPIN2SEL`, `RTC1_CFG_GPMUX1_OC1DIFFOUT`, `RTC1_CFG_GPMUX1_OC3DIFFOUT`.

MSKUV-305

Function `adi_rtc_GetInputCaptureValueEx` added to read RTC1 snapshots value triggered by Input Channel 0.

MSKUV-322

New functions have been added to help decoupling (1) setting the alarm registers from (2) synchronizing after writing the alarm registers. The added functions, `adi_rtc_SetAlarmAsync`, `adi_rtc_SetAlarmRegsAsync` and `adi_rtc_SyncAlarm`, bring the same functionality as `adi_rtc_SetAlarm` and `adi_rtc_SetAlarmRegs`, with the benefit of executing more code before waiting for synchronization.

MSKUV-328

Eliminate compilation failures for processors sharing the source with ADuCM4x50.

MSKUV-329

Mask for RTC 'wait before write'/'pend after write' bits corrected.

MSKUV-334

RTC SYNC_AFTER_WRITE macro now waits for all the monitored bits to be set, not just one. ALARM2 should also be monitored when it's written.

In addition, adi_rtc_Open cannot use disabled macros WAIT_BEFORE_WRITE and SYNC_AFTER_WRITE; the sequence of write accesses to RTC registers require the use of ALWAYS_WAIT_BEFORE_WRITE and ALWAYS_SYNC_AFTER_WRITE.

2.1.4 SPI

MSKUV-263

The SPI driver can now set the TIM bit for DMA transactions if TxBytes != 0.

2.1.5 TMR

MSKUV-294

Obsolete comments have been removed from configuration file adi_tmr_config.

2.1.6 UART

MSKUV-257

Function adi_uart_FlushRxChannel, transfer mode field is reset for the next reception buffer submission to be performed.

MSKUV-262

Function adi_uart_GetTxBuffer could block code execution because of a missing post semaphore request.

MSKUV-270

Make sure UART non-blocking transmission works fine when no callback function is registered.

MSKUV-318

Improve the way the UART driver handles FIFO timeouts.

MSKUV-335

Add support to fully drain the Rx FIFO when the Rx interrupt is being serviced.

2.1.7 Vector Table**MSKUV-261**

Added support for vector table relocation in SRAM.

3 Release Notes for ADuCM4x50 Device Family Pack 3.1.2

3.1 Differences between version 3.1.2 and prior versions

ADuCM4x50 Device Family Pack has been updated with software modifications MSKUV-289, MSKUV-290, MSKUV-291, MSKUV-292, MSKUV-293, MSKUV-300.

3.1.1 Silicon Revision

Project CMSIS Pack Component Manager has been extended with a *Silicon Revision* component which allows to select the targeted ADuCM4x50 silicon revision: 0.0. or 0.1. This helps automatically enabling silicon revision specific software parts, such as a software work around for an anomaly, e.g. software modification identified as MSKUV-290.

Defined as a variant, the *Silicon Revision* component declares a macro in the RTE_Components.h file: ADUCM4050_SI_REV, with a value reflecting the ADuCM4x50 silicon revision selected.

3.1.2 GPIO

MSKUV-300

GPIO driver API extended with `adi_gpio_GroupInterruptPolarityEnable` to determine if the interrupts are generated on the rising or falling edge of the corresponding GPIO pin.

3.1.3 PWR Driver

MSKUV-291

The following functions have been removed from the power driver

```
ADI_PWR_RESULT adi_pwr_EnableLFXTALRobustMode(const bool bEnable);  
ADI_PWR_RESULT adi_pwr_SetLFXTALRobustModeLoad(const ADI_PWR_LFXTAL_LOAD eLoad);
```

3.1.4 RTC Driver

MSKUV-289

RTC driver modified to eliminate the risk of counter overflows.

❶ Snapshots and coherent counter read accesses

Coherent counter read access is implemented using software snapshots. This means using SNAP0, SNAP1 and SNAP2 registers. As a result, there's a risk of interference between Input Capture Channel 0 and `adi_rtc_GetCount` because they both store time information in SNAP0, SNAP1 and SNAP2. This is RTC1 only.

MSKUV-290

Software work around for anomaly 2100023, an anomaly that can impact RTC registers read accesses on ADuCM4050 si. rev. 0.0 only.

This software work around

- Automatically enable when silicon revision is set to 0.0.
It can be disabled though by defining a macro in a project, `WA_21000023`, with value 0.
- Automatically disable when silicon revision is set to 0.1 since the anomaly doesn't exist in this revision.

❶ Snapshots through RTC1 Input Capture Channel 0

The work around for anomaly 21000023 - Silicon Revision 0.0 - uses SNAP0, SNAP1 and SNAP2 registers. This means using SNAP0, SNAP1 and SNAP2 registers. As a result, there's a risk of interference between Input Capture Channel 0 and `adi_rtc_GetCount` because they both store time information in SNAP0, SNAP1 and SNAP2. This is RTC1 only.

3.1.5 RTOS

MSKUV-293

The RTOS mapping has been extended with Micrium μ C/OS-II.

3.1.6 UART Driver

MSKUV-292

UART driver updated for PIO Rx transfers to support all the FIFO trigger levels. (Previous versions supported 1-byte but not 4-byte/8-byte/14-byte.)

A minor change was required in `adi_uart_SetRxFifoTriggerLevel` for this modification: the `hDevice` parameter cannot be constant anymore as the Rx FIFO trigger level must be recorded.

ADuCM4x50 DFP 3.1.2

```
ADI_UART_RESULT adi_uart_SetRxFifoTriggerLevel(  
    ADI_UART_HANDLE const hDevice,  
    ADI_UART_TRIG_LEVEL const eTriglevel  
);
```

ADuCM4x50 DFP 3.1.0

```
ADI_UART_RESULT adi_uart_SetRxFifoTriggerLevel(  
    ADI_UART_CONST_HANDLE const hDevice,  
    ADI_UART_TRIG_LEVEL const eTriglevel  
);
```

4 Release Notes for ADuCM4x50 Device Family Pack 3.1.0

4.1 Differences between version 3.1.0 and prior versions

The main changes in version 3.1.0 is the extended support for IAR Embedded Workbench.

- ADuCM4x50_DFP\3.1.0\ARM\config now includes material to fully support ADuCM4x50 in CMSIS Pack, e.g. ICF files, DDF files, flash programmer. This allows a better integration of new CMSIS packs with these resources not depending on the IAR Embedded Workbench tool kit, as in previous release.
- Source for building the flash programmer used by IAR available in ADuCM4x50_DFP\3.1.0\ARM\src\flashloader\AnalogDevices\FlashADuCM4050.
- New flash programmer.

This version now requires IAR Embedded Workbench for ARM 8.20.1 or later.

A cycle count component has been added to help evaluating the number of cycles executed by portions of code.

4.2 Required Software

4.2.1 Keil uVision

To use this ADuCM4x50 Device Family Pack with Keil uVision , you must first obtain and install:

- Keil uVision MDK v5.22 or later with ARM Compiler version 1.1.0 or later;
- Segger J-Link LITE v5.10p or later.

Install the Keil software first, then install the Segger J-Link LITE software.

4.2.2 CrossCore Embedded Studio

To use this ADuCM4x50 Device Family Pack with CrossCore Embedded Studio, you must first obtain and install:

- CrossCore Embedded Studio 2.7.0 or later.

4.2.3 IAR Embedded Workbench

To use this ADuCM4x50 Device Family Pack with IAR Embedded Workbench, you must first obtain and install:

- IAR Embedded Workbench for ARM 8.20.1 or later.

4.3 Release Testing

4.3.1 Keil uVision

This ADuCM4x50 Device Family Pack has been tested with

EZ-KIT	Emulator
ADuCM4050 LFCSP EZ-KIT version 1.0 BOM Rev 1.2	J-Link Lite
	CMSIS-DAP
EV-COG-AD4050LZ	CMSIS-DAP

4.3.2 CrossCore Embedded Studio

This ADuCM4x50 Device Family Pack has been tested with

EZ-KIT	Emulator
ADuCM4050 LFCSP EZ-KIT version 1.0 BOM Rev 1.2	ICE-2000

4.3.3 IAR Embedded Workbench

This ADuCM4x50 Device Family Pack has been tested with

EZ-KIT	Emulator
ADuCM4050 LFCSP EZ-KIT version 1.0 BOM Rev 1.2	J-Link Lite
	CMSIS-DAP
EV-COG-AD4050LZ	CMSIS-DAP

4.3.4 License Checking

Use of ADuCM4x50 Device Family Pack software is subject to the Software License Agreement presented during installation.

The details of this Software License Agreement can be found in the CMSIS pack installation directory, in AnalogDevices\ADuCM4x50_DFP\3.1.0\License.

4.4 Release Content

This release contains the following sets of components:

- Source files for the ADuCM4x50 device family drivers. These components are authored by Analog Devices, for use on the ADuCM4x50 processor.
- 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.
- Additional utilities. These components are authored by Analog Devices, and assist in the generation of applications for the ADuCM4x50 processor family.
- Documentation.

4.4.1 Source files for device family 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.

4.4.2 Location

The ADuCM4x50 Device Family Pack 3.1.0 will be installed into the CMSIS pack directory for the targeted development environment:

Keil uVision	<keil_root>\ARM\PACK\AnalogDevices\ADuCM4x50_DFP\3.1.0
---------------------	--

CCES	<cces_root>\ARM\PACK\AnalogDevices\ADuCM4x50_DFP\3.1.0
IAR Embedded Workbench	<iar_packrepo>\AnalogDevices\ADuCM4x50_DFP\3.1.0

with

- <keil_root>
 - The location where Keil uVision is installed
e.g. **C:\Keil_v5**.
- <cces_root>
 - The location where CrossCOre Embedded Studio is installed,
e.g. **C:\Analog Devices\CrossCore Embedded Studio 2.7.0**.
- <iar_packrepo>
 - The location where IAR Embedded Workbench installs CMSIS packs,
e.g. **C:\Users\<windows_username>\AppData\local\IAR Embedded Workbench\PackRepo**.

4.4.3 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.

4.4.4 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](mailto: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.

4.4.5 Examples

This ADuCM4x50 Device Family Pack comes with a very simple example which requires multiple drivers (DMA, UART, Power)

Examples for drivers

1.	HelloWorld	<ul style="list-style-type: none">• Demonstrate how to create a simple application that prints "Hello, world!".
----	------------	---

4.5 Known Issues

For the latest anomalies please consult our [Software and Tools Anomalies Search](#) page.