
# SWTE PP Board Support Packages

Version , May 2026


http://www.analog.com
processor.tools.support@analog.com

# 1 Table of Contents

# Copyright Information



# Disclaimer



# Trademark and Service Mark Notice

# 2 Introduction

Thank you for installing the ADuCM4x50 Device Family Pack (DFP). This document describes the changes for the ADuCM4x50 Device Family Pack 5.0.0. ADuCM4x50 Device Family Pack 5.0.0 is supported in Keil µVision, CrossCore Embedded Studio® (CCES) and IAR Embedded Workbench.

**This product is subject to control under the US Export Regulation. The Export Control Classification Number (ECCN) is 5D002 (unrestricted).**

# 3 Release Notes for ADuCM4x50 Device Family Pack 5.0.0

## 3.1 Differences between version 5.0.0 and prior versions.

### 3.1.1 Toolchain & CMSIS

- Pack built and tested with CCES 3.0.3, Keil MDK6 & IAR 9.70.1.
- GCC startup file updated for compatibility with ARM CMSIS Pack 6.3.0.

### 3.1.2 Debug & Tools

- Fixed OpenOCD compatibility issues.
- Updated debug logging to use `snprintf` (safer memory handling).
- Added LED blink support to the `hello_world` example
- Updated Pinmux tool with latest GPIO support.
- Updated the Flashloader to align with Keil MDK6.

### 3.1.3 Crypto

InitializeDevData() corrected to avoid clearing the entire handle structure. Runtime state, callback, transaction pointer, and computation context are cleared, while hardware registers (INTEN, STAT, CFG) are explicitly cleared.

Fixed crypto driver behavior in ECB mode by removing dependency on unused fields.

### 3.1.4 PWR

Macros and comments for CTL5 and related registers are corrected to align with the ADuCM4x50 HRM.

### 3.1.5 CRC

Improved NULL pointer management in CRC to eliminate Coverity warnings.

### 3.1.6 Silicon

Updated the .pdsc file for silicon version as 0.1 and Removed version 0.0 due to the unavailability of data for that version.

For the existing projects, follow the steps below to resolve the build issue:

1. Open the project and unselect silicon version `0.0 in the system.rteconfig file`.
2. Save the `system.rteconfig` file.
3. Navigate to **Device** → **Silicon Revision** and select silicon version `0.1`.
4. Save the project configuration.
5. built the project.

### 3.1.7 RTC

Masked interrupts issue fixed. Interrupt status registers (SR0/SR2/SR3) are read and cleared immediately in both RTC0_Int_Handler() and RTC1_Int_Handler(), ensuring newly arriving interrupts are not masked.

Corrected the bitfield in the aRTCConfig for SS1.

### 3.1.8 SVD

SVD warnings cleared by adjusting 'size' fields for FLCC0, FLCC0_CACHE, GPIO0, PMG0, XINT0, CLKG0_OSC, PMG0_TST, CLKG0_CLK, and BUSM0. Corrected baseAddress entries for PMG0_TST. Aligned register block sizes with the hardware reference manual.

### 3.1.9 FreeRTOS

FreeRTOS example projects updated to CMSIS Pack with FreeRTOS 11.2.0 and eliminated all compiler warnings.

# 4 Release Notes for ADuCM4x50 Device Family Pack 4.0.0

## 4.1 Differences between version 4.0.0 and prior versions.

### 4.1.1 ADC

Code clearing bits in STATUS register in functions DmaFIFOManage and InterruptFIFOManage improved.

### 4.1.2 Crypto

STAT register self-assignment instructions eliminated.

### 4.1.3 I2C

Rx Overflow and Tx Underflow detection fixed.

Functions adi_i2c_GetHWMaskedErrors and adi_i2c_SetHWMaskedErrors added to ignore HW errors, e.g. Rx Underflow, when desired.

Function adi_i2c_GetNumberOfDataProcessed added to get the number of bytes sent/received when a transaction is being serviced or an error occurred.

### 4.1.4 RTC

64-
bit data handling improved in adi_rtc_GetCoherentCounterValues. (Support for anomaly 21000023 in silicon revision 0.0)

Function adi_rtc_GetISOENB added to check the ISENB bit value.

### 4.1.5 UART

ACR register bits ABE, DNIEN and TOIEN are now cleared properly when calling adi_uart_EnableAutobaud to disable autobaud.

The UART DMA driven model has been extended with Ping-Pong and Scatter-Gather. This extension is disabled by default. Macro ADI_UART_DMA_EXTENSION_ENABLE must be set to 1 in the UART driver configuration file for this feature to be enabled.

Functions adi_uart_SetTxDmaMode and adi_uart_SetRxDmaMode to select PingPong or Scatter-Gather when ADI_UART_DMA_EXTENSION_ENABLE is enabled in UART configuration file.

# 5 Release Notes for ADuCM4x50 Device Family Pack 3.4.0

## 5.1 Differences between version 3.4.0 and prior versions.

### 5.1.1 DMA

API extended with two functions:

- adi_dma_Enable: enable/disable the DMA
- adi_dma_ReInit: force adi_dma_init to fully re-execute

### 5.1.2 GCC - ARM CMSIS Pack 5.6.0

GCC startup files updated to make them compatible with the introduction of `struct __copy_table_t` in ARM CMSIS Pack 5.6.0.

### 5.1.3 I2C

Fix the number of writable bytes in FIFO in functions commenceTransmit and commenceReceive.

Eliminate the risk of error caused by a semaphore posting in I2C interrupt handler.

### 5.1.4 PWR

Fix for DMA not being re-enabled when waking up from hibernation.

### 5.1.5 SPI

SPI driver improved to eliminate the risk of Tx Underflow for DMA driven transactions

### 5.1.6 RTC

Fix for RTC interrupts cleared without being serviced.

Make sure all the pending interrupts are cleared when executing afi_rtc_Open.

### 5.1.7 UART

Protect concurrent access to IEN register: IEN is updated in UART interrupt handlers, which can cause issues when the register is being modified by UART functions called by users' applications.

API extended with function adi_uart_GetRX.

# 6 Release Notes for ADuCM4x50 Device Family Pack 3.3.0

## 6.1 Differences between version 3.3.0 and prior versions.

### 6.1.1 Processor Files

Protect against risks of macro redefinitions in processor files.

### 6.1.2 System Files

File system_ADuCM4050.c updated to enable/disable bus error on CRC error by default and to enable SRAM parity by default. (Both disable by default.)

### 6.1.3 FreeRTOS

RTOS macros for critical section redefined to properly disable interrupts.

### 6.1.4 uC/OS-III

Support to disable SEM_DELETE using macro OS_CFG_SEM_DEL_EN added

### 6.1.5 ADC

ADC API extended with function adi_adc_EnableIRQ to enable/disable interrupts.

### 6.1.6 Flash Controller

Different flash controller macros, e.g. ADI_FEE_NUM_INSTANCES, FEE_FLASH_SIZE, FEE_BLOCK_SHIFT, FEE_MAX_NUM_PAGES, etc. now located in adi_flash.h

### 6.1.7 I2C

Support for bus clear operation added:

- Configuration parameters extended with ADI_I2C_CFG_MCTL_BUSCLR and ADI_I2C_CFG_MCTL_STOPBUSCLR to configure the I2C with a default behavior.
- Function adi_i2c_SetBusClear added to dynamically set/clear the BUSCLR and STOPBUSCLR bits.

Incomplete Rx Transmission detection added.

### 6.1.8 PWR

Function adi_pwr_EnableClockSource returns an error if a call to adi_gpio_InputEnable fails.

Function adi_pwr_ExitLowPowerMode clears the PWRMOD register along with bits SLEEPONEXIT and SLEEPDEEP in SCR register when exiting low power modes.

### 6.1.9 SPI

SPI DMA interrupt handling simplified and improved to handle Tx bytes number > Rx bytes number.

SPI configuration macros ADI_SPI_TRAP_RXOVR and ADI_SPI_TRAP_TXUNDR introduced to enable/disable RXOVR and TXUNDR error detection in SPI interrupt handlers. (Enable by default.)

### 6.1.10 UART

Macro guarded Rx Buffer extension to help users' callback functions to pad the Rx buffer when the number of bytes received is not a multiple of the number of bytes that triggers an interrupt.

Macro guarded Rx Buffer fast draining extension added.

Data transfer mode set to none when flushing Tx buffers.

# 7 Release Notes for ADuCM4x50 Device Family Pack 3.2.0

## 7.1 Differences between version 3.2.0 and prior versions.

### 7.1.1 ADC

Documentation for SetAcquisitionTime function improved.

### 7.1.2 Crypto

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

Allow 0-length input data for CCM mode.

### 7.1.3 RTC

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

Proper support for function adi_rtc_SetAutoReloadValue on ADuCM4x50 implemented.

Support to dynamically configure the input capture overwrite enable feature.

Function adi_rtc_SetAlarmRegs added.

Function adi_rtc_SetSensorStrobeChannelMask now preserves previous settings.

Proper RTC interrupts support for both ADuCM302x and ADuCM4x50.

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.

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

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.

Eliminate compilation failures for processors sharing the source with ADuCM4x50.

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

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.

### 7.1.4 SPI

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

### 7.1.5 TMR

Obsolete comments have been removed from configuration file adi_tmr_config.

### 7.1.6 UART

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

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

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

Improve the way the UART driver handles FIFO timeouts.

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

### 7.1.7 Vector Table

Added support for vector table relocation in SRAM.

# 8 Release Notes for ADuCM4x50 Device Family Pack 3.1.2

## 8.1 Differences between version 3.1.2 and prior versions

### 8.1.1 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.

### 8.1.2 GPIO

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.

### 8.1.3 PWR Driver

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);
```

### 8.1.4 RTC Driver

RTC driver modified to eliminate the risk of counter overflows.

> **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.

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 disable 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.

**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.

### 8.1.5 RTOS

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

### 8.1.6 UART Driver

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
                );
```

# 9 Release Notes for ADuCM4x50 Device Family Pack 3.1.0

## 9.1 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.

# 10 Required Software

## 10.1 Keil µVision

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

- Keil µVision MDK v6 or later with ARM Compiler version 6.3.0 or later;
- Segger J-Link LITE v5.10p or later.

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

## 10.2 CrossCore Embedded Studio

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

- CrossCore Embedded Studio 3.0.3 or later.

## 10.3 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 9.70.1 or later.

## 10.4 Release Testing

### 10.4.1 Keil µVision

This ADuCM4x50 Device Family Pack has been tested with

| Boards | Emulator |
|---|---|
| EV-COG-AD4050LZ | J-Link |

### 10.4.2 CrossCore Embedded Studio

This ADuCM4x50 Device Family Pack has been tested with

| Boards | Emulator |
|---|---|
| EV-COG-AD4050LZ | ICE-2000 |

### 10.4.3 IAR Embedded Workbench

This ADuCM4x50 Device Family Pack has been tested with

| Boards | Emulator |
| --- | --- |
| EV-COG-AD4050LZ | CMSIS-DAP |

### 10.4.4 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\5.0.0\License, for release 5.0.0.

## 10.5 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.

### 10.5.1 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.

### 10.5.2 Location

The ADuCM4x50 Device Family Pack 5.0.0 will be installed into the CMSIS pack directory for the targeted development environment:

| | |
| --- | --- |
| Keil µVision | <keil |

| | |
|---|---|
| | _root>\ARM\PACK\Analog Devices\ADuCM4x50_DFP\5.0 |

| | |
|---|---|
| | .0 |
| CCES | <cces_root>\ARM\PACK\Analog Devices\ADuCM4x50_ |

| | |
|---|---|
| | DFP\5.0.0 |

| | |
|---|---|
| **IAR Embedded Workbench** | <iar_packrepo>\Analog Devices\ADuCM4x50_DFP\5.0.0 |

with

- `<keil_root>`
    - The location where Keil µVision 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 3.0.3`**
- `<iar_packrepo>`
    - The location where IAR Embedded Workbench installs CMSIS packs,
      e.g. **`C:\Users\<windows_username>\IAR-CMSIS-Packs`**.

### 10.5.3 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.

### 10.5.4 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.
- 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.

### 10.5.5 Examples

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

#### Examples for drivers

| | | |
|---|---|---|
| 1. | Hello World | • Demonstrate how to create a simple application that prints "Hello, world!".<br>• Add LED code to the Hello World example to confirm execution when console output is not available with DAP debugging |

## 10.6 Known Issues

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