Software Changes from ADuCM3029 to ADuCM4050

Software Changes from ADuCM3029 to ADuCM4050
Version , March 2017

http://www.analog.com

Contents

1 Introduction

This document describes the ADuCM4050 drivers relative to the ADuCM3029 drivers. The ADuCM3029 drivers were used as a starting point for the ADuCM4050 drivers, since the hardware is very similar for most peripherals. However, the ADuCM3029 drivers were general purpose drivers and the ADuCM4050 drivers will target resource constrained Internet of Things (IoT) devices. Due to this change in intended use, the ADuCM4050 drivers have a smaller footprint, require fewer CPU cycles, and are simpler to use when compared to the ADuCM3029 drivers. For developers moving from the ADuCM3029 drivers to the ADuCM4050 drivers, a summary of the changes will be valuable. This document highlights the functional differences, API differences, and footprint reduction in the ADuCM4050 drivers when compared to the ADuCM3029 drivers.

2 Footprint Calculations

The footprint metrics provided in this document are based on the examples in the ADuCM4050 and ADuCM3029 Board Support Packages. These examples represent typical driver use cases, and the footprint metrics are taken from the map file generated by the linker. This means that the footprint metrics are calculated after the linker has eliminated any unused functions and performed any optimizations. An application using more or less of the driver functions will see variation from the footprint metrics provided here.

DMA operations are performed using a centralized DMA service in the ADuCM3029 drivers. This was changed in the ADuCM4050, and the majority of DMA code is implemented in the driver directly, with the exception of a small amount of common code used for initialization and error handling. Because of this change, the reader should note that the footprint hit due to the DMA code in the ADuCM3029 case would only be experienced once in an application using multiple drivers since this DMA code would be shared. This is also true for the very small amount of common DMA code in the ADuCM4050 case.

Reduction is calculated by dividing the ADuCM3029 footprint by the ADuCM4050 footprint. If there was no reduction or the ADuCM4050 driver is larger, no reduction is calculated.

3 Common

- DMA
  - No centralized DMA service
    - Each driver programs the DMA according to its needs
  - No ping-pong mode support
  - One common DMA file for initialization and error handling
    - Used by drivers, not by user application
  - Basic DMA mode only (one-shot)
    - 1 KB transfer of 8 or 16-bit width depending on the driver
    - Application must break up larger transfers
- Drivers vs. Services
  - No differentiation
  - Include paths are always <drivers/xxx/adi_xxx.h>
- Not all hardware features supported
  - Some only supported through static configuration
- Transaction Modes - Supported on a per peripheral basis
  - Callback
  - Blocking
  - Non-blocking
- DMA vs. PIO
  - Switched on a per transaction basis
  - Not configured through an API
- Interrupt dispatch layer removed
- No OSAL
  - RTOS calls made through macros which map to RTOS APIs
- Hardware errors
  - Reported through transaction APIs in bit-fields
- Streaming
  - Some peripherals only support single buffer

- SystemInit call not required
    - Performed in startup code
- Startup code moved to assembly file
- Project templates

4 SPI

- Single buffer only - no streaming
- All transaction modes are supported
- Both Master and Slave supported
- Read Control hardware feature supported
- Flow Control hardware feature unsupported
- DMA on a per transaction basis

ADuCM3029	ADuCM4050	Change
adi_spi_ReadWrite	adi_spi_MasterReadWrite	API name change
adi_spi_MasterTransfer	adi_spi_MasterSubmitBuffer	APIs are no longer mode based. This is always a non-blocking API.
adi_spi_SlaveTransfer	adi_spi_SlaveReadWrite	Slave mode blocking API.
adi_spi_SlaveTransfer	adi_spi_SlaveSubmitBuffer	APIs are no longer mode based. This is always a non-blocking API.
Was not there	adi_spi_GetBuffer	For either Master or Slave mode. Required to call this API when operating in non-blocking and non-callback mode.
adi_spi_MasterComplete	adi_spi_isBufferAvailable	API name change, conforms with the general driver model
adi_spi_SlaveComplete	adi_spi_isBufferAvailable	API name change, conforms with the general driver model
adi_spi_SetClockPhase	Removed	Statically configured
adi_spi_SetClockPolarity	Removed	Statically configured

adi_spi_SetFifoRxFlush	Removed	Statically configured
adi_spi_SetInterruptMode	Removed	Statically configured
adi_spi_SetLsbFirst	Removed	Statically configured
adi_spi_SetMisoOutput	Removed	Statically configured
adi_spi_SetWiredOrMode	Removed	Statically configured
adi_spi_SetThreepinMode	Removed	Statically configured
adi_spi_SetOverlapMode	Removed	Statically configured
adi_spi_SetBurstSize	Removed	Statically configured
adi_spi_SetReadySignalPolarity	Removed	Statically configured
adi_spi_SetFlowMode	Removed	Statically configured
adi_spi_WaitTime	Removed	Statically configured
adi_spi_GetChipSelect	Removed	Not required.
adi_spi_SetInterruptMode	Removed	Driver is no longer mode based. DMA or Interrupt mode are specified in the ADI_SPI_TRANSCEIVER structure.
adi_spi_SetMasterMode	Removed	Driver is no longer mode based. DMA or Interrupt mode are specified in the ADI_SPI_TRANSCEIVER structure.
adi_spi_SetTransferInitiateMode	Removed	Controlled entirely by the driver
adi_spi_SetInterrruptMask	Removed	Controlled entirely by the driver
adi_spi_SetTransmitBytes	Removed	RD_CTL configured entirely within the ADI_SPI_TRANSCEIVER structure.

adi_spi_SetReadCommand	Removed	RD_CTL configured entirely within the ADI_SPI_TRANSCEIVER structure.

Numbers obtained with ADuCM4x50_EZ_Kit\examples\spi\spi_loopback

Footprint	**ADuCM4050 (bytes)** SPI + DMA = total	**ADuCM3029 (bytes)** SPI + DMA = total	**Reduction**
RO Code (Debug)	2252 + 424 = 2676	2916 + 2704 = 5620	2.10x
RO Data (Debug)	59 + 8 = 67	30 + 312 = 342	
RW Data (Debug)	60 + 1248 = 1308	84 + 1172 = 1256	
RO Code (Release)	1392 + 224 = 1616	1624 + 1840 = 3464	2.14x
RO Data (Release)	47 + 4 = 51	30 + 52 = 82	
RW Data (Release)	60 + 1248 = 1308	84 + 1172 = 1256	

5 I2C

- Master-Mode only operation (removed Slave-Mode support).
- Interrupt (PIO) mode operation only (removed DMA support).
- Single buffer transport model (removed double-buffering).
- Consolidated and eliminated various APIs.
- No Callbacks.
- 7-Bit Addressing Mode only (removed 10-Bit Addressing Mode).
- HW Events passed via API parameters, not separate APIs.

ADuCM3029	ADuCM4050	Change
adi_i2c_Open	adi_i2c_Open	Removed Master/Slave mode parameter
adi_i2c_RegisterCallback		Eliminated (no callbacks in i2c)
adi_i2c_SubmitTxBuffer	adi_i2c_SubmitBuffer	Consolidated
adi_i2c_SubmitRxBuffer	ditto	Consolidated
adi_i2c_IsRxBufferAvailable	adi_i2c_IsBufferAvailable	Consolidated
adi_i2c_IsTxBufferAvailable	ditto	Consolidated
adi_i2c_GetRxBuffer	adi_i2c_GetBuffer	Consolidated
adi_i2c_GetTxBuffer	ditto	Consolidated
adi_i2c_Enable		Eliminated (device enable managed in driver)
adi_i2c_Write	adi_i2c_ReadWrite	Consolidated
adi_i2c_Read	ditto	Consolidated
adi_i2c_SetBitRate	adi_i2c_SetBitRate	bitrate parameter type changed to 32-bit

adi_i2c_SetDutyCycle		Eliminated (uses 50/50 duty cycle)
adi_i2c_SetHardwareAddress	adi_i2c_SetSlaveAddress	Renamed
adi_i2c_SetHWAddressWidth		Eliminated (10-bit addressing removed)
adi_i2c_EnableGeneralCall		Eliminated (slave-mode removed)
adi_i2c_SetAlternateDevID		Eliminated (General-Call target address is fixed to reserved address zero)
adi_i2c_SetAutoStretch		Eliminated (moved to static configuration)
adi_i2c_EnableDMA		Eliminated (DMA overhead removed)
adi_i2c_GetEvents		Eliminated (HW events passed via adi_i2c_ReadWrite, adi_i2c_GetBuffer, and adi_i2c_IssueGeneralCall APIs)
adi_i2c_IsEventAvailable		Eliminated (HW events indicated withADI_I2C_HW_ERROR_DETECTED API return code)

Numbers obtained with ADuCM4x50_EZ_Kit\examples\i2c\temperature_sensor

Footprint	**ADuCM4050 (bytes)**	**ADuCM3029 (bytes)**	**Reduction**
RO Code (Debug)	1616	4164	2.58x
RO Data (Debug)	28	3	
RW Data (Debug)	30	16	
RO Code (Release)	1172	2544	2.17x
RO Data (Release)	20	3	

RW Data (Release)	32	16	

6 UART

- There are no longer DMA and PIO modes. DMA or PIO usage is now decided on a buffer by buffer basis.
- There are no longer specific calls to enable the data flow on each channel.
  - Whenever there is a buffer submitted, data flow will be enabled on that channel.
  - The "Receive Buffer Full" interrupt will always be enabled to detect incoming data to allow for use of 16 byte hardware FIFO's.
- Error codes can be either returned through the callback or through a parameter in the API functions depending on the mode.
- Removed ability to statically eliminate DMA code.
- Added functionality to flush Rx and Tx channels for aborting a transaction.
- No longer a specific hardware event callback. Hardware events and buffer events now share the same callback.
- Combine configuring and enabling autobaud into one function.
  - The autobaud key character is no longer configurable. It has been hard coded to a carriage return.
- Some of the dynamic configuration functions have been removed.

ADuCM3029	ADuCM4050	Change
adi_uart_SubmitRx/TxBuffer	adi_uart_SubmitRx /TxBuffer	The bDMA flag is added to choose between PIO/DMA on per transaction basis.
adi_uart_GetRx/TxBuffer	adi_uart_GetRx/TxBuffer	Added the pHwError parameter through which the hardware errors are returned when not in callback mode.

ADuCM3029	ADuCM4050	Change
adi_uart_Read/Write	adi_uart_Read/Write	The bDMA flag is added to choose between PIO/DMA on per transaction basis. Added the pHwError parameter through which the hardware errors are returned when not in callback mode.
adi_uart_GetBaudRate	adi_uart_GetBaudRate	Added parameter return variable to detect autobaud errors when not in callback mode.
adi_uart_EnableAutobaud	adi_uart_EnableAutobaud	Added parameter option to decided if errors will be returned using a callback or through the parameter return variable
adi_uart_ClearRx/TxFifo	adi_uart_FlushRx/TxFifo	Renamed
adi_uart_ClearRxFifo	adi_tmr_GetCapturedCount	Renamed
Not Available	adi_uart_FlushRx /TxChannel	Added function to flush the Rx channel even if data is in flight. This will flush the channel as well as disable Rx interrupts.
adi_uart_EnableRx/Tx	Removed	This was removed. Whenever a buffer is active the channel will be enabled. For Rx the channel will always be enabled, unless it is flushed, which will disable Rx interrupts.
adi_uart_EnableDMAMode	Removed	No more "modes". This is all handled on a buffer by buffer basis
adi_uart_EnableDMAModeForRx/Tx	Removed	No more "modes". This is all handled on a buffer by buffer basis

ADuCM3029	ADuCM4050	Change
adi_uart_GetHWErrorStatus	Removed	Errors will now be returned either through the callback or through a parameter
adi_uart_ConfigAutobaud	Removed	This is now handled as part of adi_uart_EnableAutobaud(). The key character is no longer configurable and is hard coded to a carriage return.
adi_uart_InvertRxLine	Removed	Can be configured through static config.
adi_uart_DisableRxDuringTx	Removed	Can be configured through static config
adi_uart_HoldTxDuringRx	Removed	Can be configured through static config
adi_uart_SetDeassertHalfBit	Removed	Can be configured through static config
adi_uart_SetSoutPolarity	Removed	Can be configured through static config
adi_uart_EnableRxStatusInterrupt	Removed	Can be configured through static config
adi_uart_EnableModemStatusInterrupt	Removed	Can be configured through static config
adi_uart_RegisterHWEventCallback	Removed	There is no longer a separate callback for hardware events. This is handled as part of the buffer event callback.

Numbers obtained with ADuCM4x50_EZ_Kit\examples\uart\Loopback

Footprint	ADuCM4050 (bytes) UART + DMA = total	ADuCM3029 (bytes) UART + DMA = total	Reduction
RO Code (Debug)	3004 + 392 = 3396	4456 + 2572 = 7028	2.07x

RO Data (Debug)	300 + 8 = 308	9 + 325 = 334	
RW Data (Debug)	32 + 1248 = 1280	20 + 1172 = 1192	
RO Code (Release)	2080 + 224 = 2304	2764 + 1768 = 4532	1.97x
RO Data (Release)	8 + 4 = 12	3 + 21 = 24	
RW Data (Release)	32 + 1248 = 1280	20 + 1172 = 1192	

7 TMR

- Both timers (GP and RGB) are accessed using the same driver.
- Usual driver data model is not present.
  - Since the driver requires such a small amount of memory to be retained between function calls, it is stored internally.
  - No *adi_tmr_Open* or *adi_tmr_Close* functions.
  - No handle data structure.
  - No memory allocated by application.
- Configuration functions have been consolidated.
  - Glue driver had many *adi_tmr_SetXXX* and *adi_tmr_EnableXXX* functions.
  - Muska driver has three *adi_tmr_ConfigXXX* functions with corresponding data structures.
- *adi_tmr_IsBusy* function is omitted, all status bits are checked in internally.
- *adi_tmr_GetLoadValue* is omitted.
- *adi_tmr_Reload* function has been added, to allow application code to restart timer before it expires.
- *adi_tmr_GetXXXValue* functions are renamed to *adi_tmr_GetXXXCount.*
- Large enumeration of events has been removed - *ADI_TMR_CAPTURE_EVENTS*. User must reference the HRM for the event ID and pass it as an integer.

ADuCM3029	ADuCM4050	Change
adi_tmr_Open	adi_tmr_Init	Renamed
adi_tmr_GetCurrentValue	adi_tmr_GetCurrentCount	Renamed
adi_tmr_GetCapturedValue	adi_tmr_GetCapturedCount	Renamed
adi_tmr_Close		Removed
adi_tmr_IsBusy		Removed
adi_tmr_GetLoadValue		Removed

adi_tmr_RegisterCallback	(adi_tmr_Init)	Consolidated
adi_tmr_SetLoadValue	(adi_tmr_Config)	Consolidated
adi_tmr_SetPrescaler	(adi_tmr_Config)	Consolidated
adi_tmr_SetRunMode	(adi_tmr_Config)	Consolidated
adi_tmr_SetCountMode	(adi_tmr_Config)	Consolidated
adi_tmr_SetClockSource	(adi_tmr_Config)	Consolidated
adi_tmr_EnableReloading	(adi_tmr_Config)	Consolidated
adi_tmr_EnableSyncBypass	(adi_tmr_Config)	Consolidated
adi_tmr_SetPWMidle	(adi_tmr_ConfigPwm)	Consolidated
adi_tmr_SetPWMMode	(adi_tmr_ConfigPwm)	Consolidated
adi_tmr_SetPWMMatchValue	(adi_tmr_ConfigPwm)	Consolidated
adi_tmr_SetEventToCapture	(adi_tmr_ConfigEvent)	Consolidated
adi_tmr_EnablePrescaleReset	(adi_tmr_ConfigEvent)	Consolidated
adi_tmr_EnableEventCapture	(adi_tmr_ConfigEvent)	Consolidated

Numbers obtained with ADuCM4x50_EZ_Kit\examples\tmr\tmr_example_gp

Footprint	**ADuCM4050 (bytes)**	**ADuCM3029 (bytes)**	**Reduction**
RO Code (Debug)	788	1088	1.38x
RO Data (Debug)	20	55	
RW Data (Debug)	48	80	

RO Code (Release)	376	476	1.27x
RO Data (Release)	17	55	
RW Data (Release)	48	80	

8 WDT

- All watchdog timer configuration is done only with static configuration.
  - Since the device can only be configured once, run time configuration is not necessary.
  - No *adi_wdt_GetXXX* or *adi_wdt_SetXXX* functions for configuration parameters.
- Usual driver data model is not present.
  - Since the driver requires such a small amount of memory to be retained between function calls, it is stored internally.
  - No *adi_wdt_Open* or *adi_wdt_Close* functions.
  - No handle data structure.
  - No memory allocated by application.
- Callback function registered when calling *adi_wdt_Enable*, but only relevant if interrupt mode is chosen.
- Kick function has been renamed from *adi_wdt_ResetTimer* to *adi_wdt_Kick*.
- No *adi_wdt_GetXXX* for status bits, all status bits are checked in internally.

ADuCM3029	ADuCM4050	Change
adi_wdt_Open		Removed
adi_wdt_Close		Removed
adi_wdt_RegisterCallback	(adi_wdt_Enable)	Consolidated
adi_wdt_Enable	adi_wdt_Enable	Unchanged
adi_wdt_ResetTimer	adi_wdt_Kick	Renamed
adi_wdt_GetCount	adi_wdt_GetCount	Unchanged
adi_wdt_SetIRQMode		Removed
adi_wdt_SetLoadCount		Removed

adi_wdt_SetPrescale		Removed
adi_wdt_SetWaitMode		Removed
adi_wdt_GetEnable		Removed
adi_wdt_GetIRQMode		Removed
adi_wdt_GetPrescale		Removed
adi_wdt_GetWaitMode		Removed
adi_wdt_GetCLRISyncStatus		Removed
adi_wdt_GetCTLSyncStatus		Removed
adi_wdt_GetLDSyncStatus		Removed
adi_wdt_GetIRQPendStatus		Removed
adi_wdt_GetResetCtrl		Removed
adi_wdt_GetLockedStatus		Removed
adi_wdt_Trigger		Removed

Numbers obtained with ADuCM4x50_EZ_Kit\examples\wdt\wdt_example_interrupt

Footprint	**ADuCM4050 (bytes)**	**ADuCM3029 (bytes)**	**Reduction**
RO Code (Debug)	212	584	2.75x
RO Data (Debug)	0	10	
RW Data (Debug)	4	8	
RO Code (Release)	124	396	3.19x

RO Data (Release)	0	6	
RW Data (Release)	4	8	

Note: The WDT examples have changed significiantly from the ADuCM3029 to the ADuCM4050. However, the one WDT example provided in the ADuCM3029 Board Support Package uses the WDT in interrupt mode, so the WDT interrupt mode example in the ADuCM4050 is used as a comparison.

9 GPIO/XINT

- Seperated external interrupts from GPIO driver
- This reduced the complexity of the driver and reduced code footprint.
- XINT driver handles external interrupts

ADuCM3029	ADuCM4050	Change
adi_gpio_ResetToPowerUp		Removed
adi_gpio_EnableExIRQ		Moved to XINT driver
adi_gpio_GetGroupInterruptPins		Removed
adi_gpio_DisableExIRQ		Moved to XINT driver
adi_gpio_GetGroupInterruptPolarity		Removed
adi_gpio_ClrGroupInterruptStatus		Removed
adi_gpio_GetOutputData		Removed
adi_gpio_EnableDeviceInterrupt		Removed
adi_gpio_DisableDeviceInterrupt		Removed

Numbers obtained with ADuCM4x50_EZ_Kit\examples\gpio\LED_button_callback

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	704	1402	1.99x
RO Data (Debug)	20	156	
RW Data (Debug)	20	16	
RO Code (Release)	328	728	2.22x

RO Data (Release)	20	16	
RW Data (Release)	20	16	

Numbers obtained with ADuCM4x50_EZ_Kit\examples\xint\wakeup_button	
Footprint	**ADuCM4050 (bytes)**
RO Code (Debug)	376
RO Data (Debug)	160
RW Data (Debug)	4
RO Code (Release)	252
RO Data (Release)	0
RW Data (Release)	4

10 FLASH

- Single buffer transport model (removed double-buffering).
- Added various static configuration macros.
- HW errors passed via API parameters, not separate return codes.
- DMA code is included unconditionally, so slight hit on footprint if not using DMA.

ADuCM3029	ADuCM4050	Change
adi_fee_Open	adi_fee_Open	Unchanged
adi_fee_Close	adi_fee_Close	Unchanged
adi_fee_RegisterCallback	adi_fee_RegisterCallback	Unchanged
adi_fee_PageErase	adi_fee_PageErase	Added pHwErrors parameter
adi_fee_MassErase	adi_fee_MassErase	Added pHwErrors parameter
adi_fee_Write	adi_fee_Write	Switched to parameter block and added pHwErrors parameter
adi_fee_SubmitTxBuffer	adi_fee_SubmitBuffer	Switched to parameter block and renamed
adi_fee_IsTxBufferAvailable	adi_fee_IsBufferAvailable	Renamed
adi_fee_GetTxBuffer	adi_fee_GetBuffer	Renamed and added pHwErrors parameter
adi_fee_GetPageNumber	adi_fee_GetPageNumber	Unchanged
adi_fee_GetBlockNumber	adi_fee_GetBlockNumber	Unchanged
adi_fee_VerifySignature	adi_fee_VerifySignature	Renamed start/end parameters and added pHwErrors parameter
adi_fee_WriteProtectBlock	adi_fee_WriteProtectBlock	Unchanged
adi_fee_Sleep	adi_fee_Sleep	Unchanged

ADuCM3029	ADuCM4050	Change
adi_fee_Abort	adi_fee_Abort	Unchanged
adi_fee_GetAbortAddr	adi_fee_GetAbortAddr	Unchanged
adi_fee_ConfigECC	adi_fee_ConfigECC	Unchanged
adi_fee_EnableECC	adi_fee_EnableECC	Unchanged
adi_fee_ConfigECCEvents	adi_fee_ConfigECCEvents	Unchanged
adi_fee_GetECCErrAddr	adi_fee_GetECCErrAddr	Unchanged
adi_fee_GetECCCorrections	adi_fee_GetECCCorrections	Unchanged
adi_fee_EnableDMA		Eliminated (changed from mode to transaction parameter)

Numbers obtained with ADuCM4x50_EZ_Kit\examples\flash\flash_page_write

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	2000	4636	2.3x
RO Data (Debug)	14	512	
RW Data (Debug)	1288	1180	
RO Code (Release)	1290	2844	2.2x
RO Data (Release)	10	32	
RW Data (Release)	1288	1180	

11 ADC

- API unchanged from ADuCM3029.
- No dynamic transitions between DMA mode and PIO mode.
- Default transfer is DMA, unless single transfers of single channels reads are specified by the user.

Numbers obtained with ADuCM4x50_EZ_Kit\examples\adc\adc_channel_read

Footprint	**ADuCM4050 (bytes)** ADC + DMA = Total	**ADuCM3029 (bytes)** ADC + DMA = Total	**Reduction**
RO Code (Debug)	2080 + 284 = 2364	2972 + 2776 = 5744	2.43x
RO Data (Debug)	2 + 8 = 10	202 + 297 = 499	
RW Data (Debug)	8 + 1248 = 1256	8 + 1172 = 1180	
RO Code (Release)	1504 + 184 = 1688	2140 + 1784 = 3924	2.32x
RO Data (Release)	2 + 4 = 6	2 + 37 = 39	
RW Data (Release)	8 + 1248 = 1256	8 + 1172 = 1180	

12 CRC

- API largely unchanged, aside to the removal of *adi_crc_EnableDmaMode*.
- Supports core driven (PIO) and DMA modes of operation. Mode of operation selected statically, similar to the ADuCM3029.
- Table of macro differences is also included below.

ADuCM3029	ADuCM4050	Change
adi_crc_Open	adi_crc_Open	Unchanged
adi_crc_Close	adi_crc_Close	Unchanged
adi_crc_RegisterCallback	adi_crc_RegisterCallback	Unchanged
adi_crc_EnableDmaMode		Removed
adi_crc_SetPolynomialVal	adi_crc_SetPolynomialVal	Unchanged
adi_crc_Compute	adi_crc_Compute	Unchanged
adi_crc_IsCrcInProgress	adi_crc_IsCrcInProgress	Unchanged
adi_crc_GetFinalCrcVal	adi_crc_GetFinalCrcVal	Unchanged
adi_crc_GetCurrentCrcVal	adi_crc_GetCurrentCrcVal	Unchanged
adi_crc_SetBitMirroring	adi_crc_SetBitMirroring	Unchanged
adi_crc_SetByteMirroring	adi_crc_SetByteMirroring	Unchanged
adi_crc_EnableWordSwap	adi_crc_EnableWordSwap	Unchanged
adi_crc_SetCrcSeedVal	adi_crc_SetCrcSeedVal	Unchanged
adi_crc_SetLSBFirst	adi_crc_SetLSBFirst	Unchanged

ADuCM3029	ADuCM4050	Change
ADI_CFG_CRC_ENABLE_BYTE_MIRRORING	ADI_CFG_CRC_ENABLE_BYTE_ MIRRORING	Unchanged
ADI_CFG_CRC_ENABLE_BIT_MIRRORING	ADI_CFG_CRC_ENABLE_BIT_ MIRRORING	Unchanged
ADI_CFG_CRC_SEED_VALUE	ADI_CFG_CRC_SEED_VALUE	Unchanged
ADI_CFG_CRC_POLYNOMIAL	ADI_CFG_CRC_POLYNOMIAL	Unchanged
	ADI_CFG_CRC_SOFTWARE_DMA_ CHANNEL_ID	New macro defining the DMA channel to be used in DMA driven operations. It's defaulted to software DMAchannel 7.

Numbers obtained with ADuCM4x50_EZ_Kit\examples\crc\core_driven_CRC

Core-driven	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	728	752	1.03x
RO Data (Debug)	8	7	0.87x
RW Data (Debug)	8	8	1.00x
RO Code (Release)	320	362	1.13x
RO Data (Release)	8	7	0.87x
RW Data (Release)	8	8	1.00x

Numbers obtained with ADuCM4x50_EZ_Kit\examples\crc\dma_driven_CRC_without_callback

Footprint	**ADuCM4050 (bytes)** CRC + DMA = total	**ADuCM3029 (bytes)** CRC + DMA = total	**Reduction**
RO Code (Debug)	1432 + 392 = 1824	620 + 1984 = 2604	1.43x
RO Data (Debug)	8 + 8 = 16	7 + 404 = 411	
RW Data (Debug)	8 + 1248 = 1256	8 + 1172 = 1180	
RO Code (Release)	660 + 224 = 884	244 + 1488 = 1732	1.96x
RO Data (Release)	8 + 4 = 12	7 + 139 = 146	
RW Data (Release)	8 + 1248 = 1256	8 + 1172 = 1180	

Numbers obtained with ADuCM4x50_EZ_Kit\examples\crc\dma_driven_CRC_with_callback

Footprint	**ADuCM4050 (bytes)** CRC + DMA = Total	**ADuCM3029 (bytes)** CRC + DMA = Total	**Reduction**
RO Code (Debug)	1440 + 392 = 1832	608 + 1984 = 2592	1.41x
RO Data (Debug)	8 + 8 = 16	7 + 404 = 411	
RW Data (Debug)	8 + 1248 = 1256	8 + 1172 = 1180	
RO Code (Release)	664 + 224 = 888	236 + 1488 = 1724	1.94x
RO Data (Release)	8 + 4 = 12	7 + 138 = 145	
RW Data (Release)	8 + 1248 = 1256	8 + = 1172 = 1180	

13 RTC

- API largely unchanged.
- Two APIs have been removed as interrupt status is handled privately by the driver.
- The footprint has increased slightly due to an increase in hardware features.
- The ADI_RTC_OUTPUT_CHANNEL enumeration is changed to ADI_RTC_SS_CHANNEL

ADuCM3029	ADuCM4050	Change
adi_rtc_GetInterruptStatus		Removed
adi_rtc_ClearInterruptStatus		Removed
adi_rtc_SetOutputChannelMask	adi_rtc_SetSensorStrobeChannelMask	Renamed
adi_rtc_SetOutputCompareValue	adi_rtc_SetSensorStrobeValue	Renamed
adi_rtc_GetOutputCompareValue	adi_rtc_GetSensorStrobeValue	Renamed
...	...	All other APIs unchanged

Numbers obtained with ADuCM4x50_EZ_Kit\examples\rtc\Rtc_alarm

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	2428	2132	None
RO Data (Debug)	110	6	
RW Data (Debug)	356	24	
RO Code (Release)	1648	1284	None
RO Data (Release)	110	6	
RW Data (Release)	356	24	

14 RNG

- API unchanged from ADuCM3029.
- Default driver configuration has been changed so the device can be used out of the box without modifications.
    - In the ADuCM3029, applications either had to change the configuration file or call APIs to change the parameters dynamically.

ADuCM3029	ADuCM4050	Change
adi_rng_Open	adi_rng_Open	Unchanged
adi_rng_Close	adi_rng_Close	Unchanged
adi_rng_Enable	adi_rng_Enable	Unchanged
adi_rng_EnableBuffering	adi_rng_EnableBuffering	Unchanged
adi_rng_SetSampleLen	adi_rng_SetSampleLen	Unchanged
adi_rng_GetRdyStatus	adi_rng_GetRdyStatus	Unchanged
adi_rng_GetStuckStatus	adi_rng_GetStuckStatus	Unchanged
adi_rng_GetRngData	adi_rng_GetRngData	Unchanged
adi_rng_GetOscCount	adi_rng_GetOscCount	Unchanged
adi_rng_GetOscDiff	adi_rng_GetOscDiff	Unchanged
adi_rng_RegisterCallback	adi_rng_RegisterCallback	Unchanged
adi_rng_GetSampleLen	adi_rng_GetSampleLen	Unchanged

Numbers obtained with ADuCM4x50_EZ_Kit\examples\rng\rng_example (RNG_ENABLE_CALLBACK = 0)			
Footprint	**ADuCM4050 (bytes)**	**ADuCM3029 (bytes)**	**Reduction**

RO Code (Debug)	540	712	1.3x
RO Data (Debug)	8	8	
RW Data (Debug)	8	8	
RO Code (Release)	304	412	1.3x
RO Data (Release)	8	2	
RW Data (Release)	8	8	

Numbers obtained with ADuCM4x50_EZ_Kit\examples\rng\rng_example (RNG_ENABLE_CALLBACK = 1)

Footprint	**ADuCM4050 (bytes)**	**ADuCM3029 (bytes)**	**Reduction**
RO Code (Debug)	576	756	1.3x
RO Data (Debug)	8	8	
RW Data (Debug)	8	8	
RO Code (Release)	334	440	1.3x
RO Data (Release)	8	2	
RW Data (Release)	8	8	

15 BEEP

- Significant API change.
- More emphasis on being able to play notes or sequences rather than controlling specific registers and managing callbacks.
- New API: *adi_beep_Wait* which allows the application to block until the note or sequence is complete.
- The new driver allows a reduction in code at the application level, with only a marginal increase within the driver.

ADuCM3029	ADuCM4050	Change
adi_beep_Open	adi_beep_Open	Callback parameters have been removed, and replaced with a separate API.
adi_beep_Close	adi_beep_Close	Unchanged
	adi_beep_RegisterCallback	New API to replace the removed parameters in adi_beep_Open().
adi_beep_ConfigInterruptMask		Removed API
adi_beep_GetInterruptMask		Removed API
adi_beep_ToneEnable		Removed API
adi_beep_SetTone		Removed API
adi_beep_GetTone		Removed API
adi_beep_SetSequence		Removed API
adi_beep_Enable	adi_beep_Enable	Unchanged
adi_beep_GetBusyStatus	adi_beep_GetBusyStatus	Unchanged
	adi_beep_PlayNote	New API to play a single note.

ADuCM3029	ADuCM4050	Change
	adi_beep_PlayTwoTone	New API to play a repeating two-tone sequence.
	adi_beep_PlaySequence	New API to play any sequence of notes.
	adi_beep_Wait	New API to be used with the above (new) APIs to ensure that the beep sequence/note has finished. This function will block until complete.

Numbers obtained with ADuCM4x50_EZ_Kit\examples\beep\beeper_example

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	984	864	None
RO Data (Debug)	116	140	
RW Data (Debug)	8	8	
RO Code (Release)	576	536	None
RO Data (Release)	0	5	
RW Data (Release)	8	8	

16 ADXL363

- Very similar API to ADuCM3029, but a little smaller.
- Footprint figures do not include the additional savings from using new GPIO and SPI drivers.
- New API to extract data from the on-chip FIFO, and to register a callback.

ADuCM3029	ADuCM4050	Change
adi_adxl363_Open	adi_adxl363_Open	Removed callback parameters.
	adi_adxl363_RegisterCallback	New API, to replace removed callback parameters in adi_adxl363_Open()
	adi_adxl363_ReadFIFO	Added specific API to extract data from the FIFO.
...	...	All other APIs unchanged

Numbers obtained with ADuCM4x50_EZ_Kit\examples\adxl363\accel\accelerometer

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	1292	1392	1.08x
RO Data (Debug)	32	32	
RW Data (Debug)	8	8	
RO Code (Release)	912	1004	1.10x
RO Data (Release)	0	0	
RW Data (Release)	8	8	

17 CRYPTO

- PIO and DMA modes supported.
- Single buffer transport model.
- Added various static configuration macros.
- DMA code is implemented explicitly.

ADuCM3029	ADuCM4050	Change
adi_crypto_Open	adi_crypto_Open	Unchanged
adi_crytpo_Close	adi_crypto_Close	Unchanged
adi_crypto_RegisterCallback	adi_crypto_RegisterCallback	Unchanged
adi_crypto_Enable	adi_crypto_Enable	Unchanged
adi_crypto_EnableDMAMode	adi_crypto_EnableDMAMode	Unchanged
adi_crypto_SubmitXXXBuffer	adi_crypto_SubmitBuffer	Consolidated
adi_crypto_GetXXXBuffer	adi_crypto_GetBuffer	Consolidated
adi_crypto_IsXXXBufferAvailable	adi_crypto_IsBufferAvailable	Consolidated

Numbers obtained with ADuCM4x50_EZ_Kit\examples\crypto\crypto_example			
Footprint	**ADuCM4050 (bytes)** CRYPTO + DMA = total	**ADuCM3029 (bytes)** CRYPTO + DMA = total	**Reduction**
RO Code (Debug)	2796 + 392 = 3188	4224 + 2856 = 7080	2.22x
RO Data (Debug)	112 + 12 = 124	480 + 356 = 836	
RW Data (Debug)	16 + 1248 = 1264	8 + 1172 = 1180	

RO Code (Release)	1892 + 224 = 2116	2916 + 1900 = 4816	2.27
RO Data (Release)	5 + 8 = 13	1 + 24 = 25	
RW Data (Release)	16 + 1248 = 1264	8 + 1172 = 1180	

18 SPORT

- Supports core driven (PIO) and DMA modes of operation. Selected dynamically with an extra boolean parameter in *adi_sport_SubmitBuffer*.
- Supports two buffers for ping-pong mode of operation.
- Buffers can be mixed with core driven and DMA.
- Table of macro differences is also included below.

ADuCM3029	ADuCM4050	Change
adi_sport_Open	adi_sport_Open	Unchanged
adi_sport_Close	adi_sport_Close	Unchanged
adi_sport_Enable		Removed
adi_sport_SubmitBuffer	adi_sport_SubmitBuffer	Extra parameter added to select DMA or core driven operations
adi_sport_GetBuffer	adi_sport_GetBuffer	Extra parameter added to get SPORT HW events
adi_sport_IsBufferAvailable	adi_sport_IsBufferAvailable	Unchanged
adi_sport_RegisterCallback	adi_sport_RegisterCallback	Unchanged
adi_sport_EnableDmaMode		Removed
adi_sport_ConfigData	adi_sport_ConfigData	Unchanged
adi_sport_ConfigClock	adi_sport_ConfigClock	Unchanged
adi_sport_ConfigFrameSync	adi_sport_ConfigFrameSync	Unchanged
adi_sport_GetHWErrorStatus		Removed
adi_sport_SetRegister		Removed

ADuCM3029	ADuCM4050	Change
adi_sport_GetRegister		Removed
adi_sport_MultiplexSportSignal		Unchanged
adi_sport_ConfigTimerMode		Unchanged

ADuCM3029	ADuCM4050	Change
ADI_SPORT0A_ACTIVE_LOW_FS	ADI_CFG_SPORT0A_ACTIVE_LOW_FS	Renamed
ADI_SPORT0A_CLOCK_DIVISOR	ADI_CFG_SPORT0A_CLOCK_DIVISOR	Renamed
ADI_SPORT0A_CLOCK_EDGE	ADI_CFG_SPORT0A_CLOCK_EDGE	Renamed
ADI_SPORT0A_CONVT_FS_ DURATION	ADI_CFG_SPORT0A_CONVT_FS_ DURATION	Renamed
ADI_SPORT0A_CONVT_POLARITY	ADI_CFG_SPORT0A_CONVT_POLARITY	Renamed
ADI_SPORT0A_CONVT_WIDTH	ADI_CFG_SPORT0A_CONVT_WIDTH	Renamed
ADI_SPORT0A_DATA_ INDEPENDENT_FS	ADI_CFG_SPORT0A_DATA_ INDEPENDENT_FS	Renamed
ADI_SPORT0A_ENABLE_CKMUXSEL	ADI_CFG_SPORT0A_ENABLE_CKMUXSEL	Renamed
ADI_SPORT0A_ENABLE_FSMUXSEL	ADI_CFG_SPORT0A_ENABLE_FSMUXSEL	Renamed
ADI_SPORT0A_ENABLE_PACKING	ADI_CFG_SPORT0A_ENABLE_PACKING	Renamed
ADI_SPORT0A_FS_DIVISOR	ADI_CFG_SPORT0A_FS_DIVISOR	Renamed
ADI_SPORT0A_FS_ERROR_ OPERATION	ADI_CFG_SPORT0A_FS_ERROR_ OPERATION	Renamed
ADI_SPORT0A_FS_REQUIRED	ADI_CFG_SPORT0A_FS_REQUIRED	Renamed

ADuCM3029	ADuCM4050	Change
ADI_SPORT0A_GATED_CLOCK	ADI_CFG_SPORT0A_GATED_CLOCK	Renamed
ADI_SPORT0A_INTERNAL_CLK	ADI_CFG_SPORT0A_INTERNAL_CLK	Renamed
ADI_SPORT0A_INTERNAL_FS	ADI_CFG_SPORT0A_INTERNAL_FS	Renamed
ADI_SPORT0A_LATE_FS	ADI_CFG_SPORT0A_LATE_FS	Renamed
ADI_SPORT0A_LSB_FIRST	ADI_CFG_SPORT0A_LSB_FIRST	Renamed
ADI_SPORT0A_OPERATION_MODE	ADI_CFG_SPORT0A_OPERATION_MODE	Renamed
ADI_SPORT0A_SERIAL_WLEN	ADI_CFG_SPORT0A_SERIAL_WLEN	Renamed
ADI_SPORT0B_ACTIVE_LOW_FS	ADI_CFG_SPORT0B_ACTIVE_LOW_FS	Renamed
ADI_SPORT0B_CLOCK_DIVISOR	ADI_CFG_SPORT0B_CLOCK_DIVISOR	Renamed
ADI_SPORT0B_CLOCK_EDGE	ADI_CFG_SPORT0B_CLOCK_EDGE	Renamed
ADI_SPORT0B_CONVT_ FS_DURATION	ADI_CFG_SPORT0B_CONVT_ FS_DURATION	Renamed
ADI_SPORT0B_CONVT_POLARITY	ADI_CFG_SPORT0B_CONVT_POLARITY	Renamed
ADI_SPORT0B_CONVT_WIDTH	ADI_CFG_SPORT0B_CONVT_WIDTH	Renamed
ADI_SPORT0B_DATA_ INDEPENDENT_FS	ADI_CFG_SPORT0B_DATA_ INDEPENDENT_FS	Renamed
ADI_SPORT0B_ENABLE_PACKING	ADI_CFG_SPORT0B_ENABLE_PACKING	Renamed
ADI_SPORT0B_FS_DIVISOR	ADI_CFG_SPORT0B_FS_DIVISOR	Renamed
ADI_SPORT0B_FS_ERROR_ OPERATION	ADI_CFG_SPORT0B_FS_ERROR_ OPERATION	Renamed

ADuCM3029	ADuCM4050	Change
ADI_SPORT0B_FS_REQUIRED	ADI_CFG_SPORT0B_FS_REQUIRED	Renamed
ADI_SPORT0B_GATED_CLOCK	ADI_CFG_SPORT0B_GATED_CLOCK	Renamed
ADI_SPORT0B_INTERNAL_CLK	ADI_CFG_SPORT0B_INTERNAL_CLK	Renamed
ADI_SPORT0B_INTERNAL_FS	ADI_CFG_SPORT0B_INTERNAL_FS	Renamed
ADI_SPORT0B_LATE_FS	ADI_CFG_SPORT0B_LATE_FS	Renamed
ADI_SPORT0B_LSB_FIRST	ADI_CFG_SPORT0B_LSB_FIRST	Renamed
ADI_SPORT0B_OPERATION_MODE	ADI_CFG_SPORT0B_OPERATION_MODE	Renamed
ADI_SPORT0B_SERIAL_WLEN	ADI_CFG_SPORT0B_SERIAL_WLEN	Renamed
ADI_SPORTA_CFG_WORD_INC	Optimal value now automatically calculated by the driver	Removed
ADI_SPORTA_CFG_WORD_WIDTH	Optimal value now automatically calculated by the driver	Removed
ADI_SPORTB_CFG_WORD_INC	Optimal value now automatically calculated by the driver	Removed
ADI_SPORTB_CFG_WORD_WIDTH	Optimal value now automatically calculated by the driver	Removed
ADI_SPORT_CFG_ENABLE_DMA	Dynamically through adi_sport_SubmitBuffer	Removed
ADI_SPORT_CFG_ENABLE_ DMA_SUPPORT	Always available	Removed
ADI_SPORT_CFG_ENABLE_	Always available along	Removed

ADuCM3029	ADuCM4050	Change
STATIC_CONFIG_SUPPORT	dynamic configuration	

Numbers obtained with ADuCM4x50_EZ_Kit\examples\sport\loopback_int

Footprint	ADuCM4050 (bytes)	ADuCM3029 (bytes)	Reduction
RO Code (Debug)	2734	2372	None
RO Data (Debug)	883	14	
RW Data (Debug)	64	48	
RO Code (Release)	1392	1424	None
RO Data (Release)	31	14	
RW Data (Release)	64	48	

Numbers obtained with ADuCM4x50_EZ_Kit\examples\sport\loopback_dma

Footprint	ADuCM4050 (bytes) CRC + DMA = total	ADuCM3029 (bytes) CRC + DMA = total	Reduction
RO Code (Debug)	2734 + 392 = 3126	2356 + 2704 = 5060	1.62x
RO Data (Debug)	883 + 8 = 891	46 + 303 = 349	
RW Data (Debug)	64 + 1248 = 1312	48 + 1172 = 1220	
RO Code (Release)	1392 + 224 = 1616	1456 + 1840 = 3296	2.04x
RO Data (Release)	31 + 4 = 35	14 + 43 = 57	
RW Data (Release)	64 + 1248 = 1312	48 + 1172 = 1220	

Note: The ADuCM3029 Board Support Package has a single example for both interrupt and DMA driven loopback. The ADuCM3029 numbers are obtained from this single example with DMA enabled or disabled through static configuration.