Integration Manual –<Name of component>

Table of Contents

[1 Dependencies 2](#_Toc384985685)

[1.1 SWCs 2](#_Toc384985686)

[1.2 Global Functions(Non RTE) to be provided to Integration Project 2](#_Toc384985687)

[2 Configuration 3](#_Toc384985688)

[2.1 Build Time Config 3](#_Toc384985689)

[2.2 Configuration Files to be provided by Integration Project 3](#_Toc384985690)

[2.2.1 Da Vinci Parameter Configuration Changes 3](#_Toc384985691)

[2.2.2 DaVinci Interrupt Configuration Changes 3](#_Toc384985692)

[2.2.3 Manual Configuration Changes 3](#_Toc384985693)

[3 Integration 6](#_Toc384985694)

[3.1 Required Global Data Inputs 6](#_Toc384985695)

[3.2 Required Global Data Outputs 6](#_Toc384985696)

[3.3 Specific Include Path present 6](#_Toc384985697)

[4 Runnable Scheduling 7](#_Toc384985698)

[5 Memory Mapping 8](#_Toc384985699)

[5.1 Mapping 8](#_Toc384985700)

[5.2 Usage 8](#_Toc384985701)

[5.3 Non RTE NvM Blocks 8](#_Toc384985702)

[5.4 RTE NvM Blocks 8](#_Toc384985703)

[6 Compiler Settings 8](#_Toc384985704)

[6.1 Preprocessor MACRO 8](#_Toc384985705)

[6.2 Optimization Settings 8](#_Toc384985706)

[7 Revision Control Log 9](#_Toc384985707)

# Dependencies

## SWCs

|  |  |
| --- | --- |
| Module | Required Feature |
| IoHwAbsUsr | Parts of the Adc FDDs (33C or 33E) are intended to be implemented at the integration level and are typically included in IoHwAbsUsr. Note that this includes implementation of NTC 0x32 and/or NTC 0x33 as needed for the specific application.  NOTE that as of FDD33C rev 008 and FDD33E rev 002, DMA-related updates are not included in the FDDs. The looping on group busy and associated timeouts and NTC setting should not be implemented when using with DMA; alternate means of getting the data when ready and associated fault setting are outlined in the DMA FDD ES-52. |

Note : Referencing the external components should be avoided in most cases. Only in unavoidable circumstance external components should be refered. Developer should track the references.

## Global Functions(Non RTE) to be provided to Integration Project

* Adc\_Init() NOTE this is a macro mapped to Adc\_Init\_FixedCfg for Autosar interface compatibility. The Adc\_Init macro takes a parameter which is not used by the function.
* FUNC(void, ADC\_CODE) Adc\_StartGroupConversion(Adc\_GroupType Group)
* FUNC(Std\_ReturnType, ADC\_CODE) Adc\_ReadGroup(Adc\_GroupType Group, Adc\_ValueGroupRefType DataBufferPtr)
* FUNC(Adc\_StatusType, ADC\_CODE) Adc\_GetGroupStatus(Adc\_GroupType Group)
* inline uint16 Adc2\_ReadConversion(uint16 ConvId) NOTE this function directly accesses Adc RAM and should not be used when using DMA to transfer Adc data
* void Adc2\_Init1(void)
* void Adc2\_StartGroupConversion(uint8 group)

# Configuration

## Build Time Config

|  |  |  |
| --- | --- | --- |
| Modules | Notes |  |
| None |  |  |

## Configuration Files to be provided by Integration Project

Adc\_Cfg.h and Adc2\_Cfg.h. Configuration file templates are in the Tools folder.

NOTE:

For Projects using 33E, make sure “D\_ADC1CURRENTMODE\_ULS\_LGC” is defined in Adc\_Cfg.h file.

For Projects using 33C, make sure “D\_ADC1CURRENTMODE\_ULS\_LGC” is **NOT** defined in Adc\_Cfg.h file.

### Da Vinci Parameter Configuration Changes

|  |  |  |
| --- | --- | --- |
| Parameter | Notes | SWC |
| <Configurator Changes for parameters> |  |  |

### DaVinci Interrupt Configuration Changes

|  |  |  |  |
| --- | --- | --- | --- |
| ISR Name | VIM # | Priority Dependency | Notes |
| <Configurator Changes for Interrupts> |  |  |  |

### Manual Configuration Changes

| Constant | Notes | SWC |
| --- | --- | --- |
| D\_ADC1GEVTSRC\_CNT\_U32\* | Initialization value for adcREG1->G0SRC | Adc\_Cfg.h |
| D\_ADC1GEVTDMACR\_CNT\_U32\* | Initialization value for adcREG1->G0DMACR | Adc\_Cfg.h |
| D\_ADC1G1SRC\_CNT\_U32\* | Initialization value for adcREG1->G1SRC | Adc\_Cfg.h |
| D\_ADC1G1DMACR\_CNT\_U32\* | Initialization value for adcREG1->G1DMACR | Adc\_Cfg.h |
| D\_ADC1G2SRC\_CNT\_U32\* | Initialization value for adcREG1->G2SRC | Adc\_Cfg.h |
| D\_ADC1G2DMACR\_CNT\_U32\* | Initialization value for adcREG1->G2DMACR | Adc\_Cfg.h |
| D\_ADC1USEDMA\_CNT\_LGC\* | STD\_ON if using DMA to transfer ADC1 data; otherwise STD\_OFF | Adc\_Cfg.h |
| D\_HWTRGADC1GEVT\_CNT\_LGC\* | STD\_ON to reconfigure ADC1 event group for hardware trigger after initial reads; otherwise STD\_OFF | Adc\_Cfg.h |
| D\_HWTRGADC1G1\_CNT\_LGC\* | STD\_ON to reconfigure ADC1 group 1 for hardware trigger after initial reads; otherwise STD\_OFF | Adc\_Cfg.h |
| D\_HWTRGADC1G2\_CNT\_LGC\* | STD\_ON to reconfigure ADC1 group 2 for hardware trigger after initial reads; otherwise STD\_OFF | Adc\_Cfg.h |
| D\_ADC2EVINTENA\_CNT\_U32\* | Initialization value for adcREG2-> GxINTENA[D\_GROUPEV\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2EVSAMPDISEN\_CNT\_U32\* | Initialization value for adcREG2-> G0SAMPDISEN | Adc2\_Cfg.h |
| D\_ADC2EVFIFORESETCR\_CNT\_U32\* | Initialization value for adcREG2-> GxFIFORESETCR[D\_GROUPEV\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2EVDMACR\_CNT\_U32\* | Initialization value for adcREG2-> G0DMACR | Adc2\_Cfg.h |
| D\_ADC2G1INTENA\_CNT\_U32\* | Initialization value for adcREG2-> GxINTENA[D\_GROUP1\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2G1SAMPDISEN\_CNT\_U32\* | Initialization value for adcREG2-> G0SAMPDISEN | Adc2\_Cfg.h |
| D\_ADC2G1FIFORESETCR\_CNT\_U32\* | Initialization value for adcREG2-> GxFIFORESETCR[D\_GROUP1\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2G1DMACR\_CNT\_U32\* | Initialization value for adcREG2-> G0DMACR | Adc2\_Cfg.h |
| D\_ADC2G2INTENA\_CNT\_U32\* | Initialization value for adcREG2-> GxINTENA[D\_GROUP2\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2G2SAMPDISEN\_CNT\_U32\* | Initialization value for adcREG2-> G0SAMPDISEN | Adc2\_Cfg.h |
| D\_ADC2G2FIFORESETCR\_CNT\_U32\* | Initialization value for adcREG2-> GxFIFORESETCR[D\_GROUP2\_CNT\_U8] | Adc2\_Cfg.h |
| D\_ADC2G2DMACR\_CNT\_U32\* | Initialization value for adcREG2-> G0DMACR | Adc2\_Cfg.h |
| D\_HWTRGADC2GEVT\_CNT\_LGC\* | STD\_ON to reconfigure ADC2 event group for hardware trigger after initial reads; otherwise STD\_OFF | Adc2\_Cfg.h |
| D\_HWTRGADC2G1\_CNT\_LGC\* | STD\_ON to reconfigure ADC2 group 1 for hardware trigger after initial reads; otherwise STD\_OFF | Adc2\_Cfg.h |
| D\_HWTRGADC2G2\_CNT\_LGC\* | STD\_ON to reconfigure ADC2 group 2 for hardware trigger after initial reads; otherwise STD\_OFF | Adc2\_Cfg.h |
| D\_ADC2USEDMA\_CNT\_LGC\* | STD\_ON if using DMA to transfer ADC2 data; otherwise STD\_OFF | Adc2\_Cfg.h |

\*NOTE: See template Adc\_Cfg.h and Adc2\_Cfg.h files in Tools folder for settings to maintain pre-DMA (FDD33C rev008 and FDD33E rev002) behavior.

\*\*NOTE: Additional manual configuration constants are needed, and are included in the template config header files, but are not listed here. To be added in CR 11738 complete design review of this component.

# Integration

## Required Global Data Inputs

None

## Required Global Data Outputs

ADC1OffsetComp\_Cnt\_u8p8

ADC2OffsetComp\_Cnt\_u8p8 (From ADC2 to Current Measurement)

## Specific Include Path present

Yes

# Runnable Scheduling

This section specifies the required runnable scheduling.

|  |  |  |
| --- | --- | --- |
| Init | Scheduling Requirements | Trigger |
| Adc\_Init() | Before **Adc2\_Init1**  Before **IoHwAb\_Init** After **SystemTime\_Init** After **Dma\_Init** | ECU Startup |
| Adc2\_Init1() | Before **PWMCdd\_Init**  Before enabling Motor Control ISR  After **SystemTime\_Init** After **Dma\_Init** | ECU Startup |

\*NOTE: Depending on configuration, some of the other listed init functions may not be present. Other application-specific scheduling requirements may exist.

|  |  |  |
| --- | --- | --- |
| Runnable | Scheduling Requirements | Trigger |
| Adc\_StartGroupConversion | As determined by application needs and configuration |  |
| Adc2\_StartGroupConversion | As determined by application needs and configuration |  |

**.**

# Memory Mapping

## Mapping

|  |  |  |
| --- | --- | --- |
| Memory Section | Contents | Notes |
| ADC2\_START\_SEC\_CODE |  |  |
| ADC2\_START\_SEC\_CONST\_32 |  |  |
| ADC\_START\_SEC\_CONST\_32 |  |  |
| ADC\_START\_SEC\_CODE |  |  |

\* Each …START\_SEC… constant is terminated by a …STOP\_SEC… constant as specified in the AUTOSAR Memory Mapping requirements.

## Usage

|  |  |  |
| --- | --- | --- |
| Feature | RAM | ROM |
| <Memmap usuage info> |  |  |

Table 1: ARM Cortex R4 Memory Usage

## Non RTE NvM Blocks

|  |
| --- |
| Block Name |
| <NVM block used Non RTE functions > |

Note : Size of the NVM block if configured in developer

## RTE NvM Blocks

|  |
| --- |
| Block Name |
| <NVM block used in RTE functions > |

Note : Size of the NVM block if configured in developer

# Compiler Settings

## Preprocessor MACRO

<Define all the preprocessor Macros needed and conditions when needed>.

## Optimization Settings

<Define Optimization levels that are needed and conditions when needed>.

# Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author** |
| 1 | Initial version | 09-Apr-13 | Selva |
| 2 | Updated to latest integration manual template. Updated per changes made for use with DMA. Added init function scheduling requirements that had previously been listed in the MDDs. | 11-Apr-14 | KMC |
| 3 | Updated for ADC offset compensation | 27-Jun-14 | Selva |