# Module -- Adc Core

# High-Level Description

The Adc Common module provides “stateless” application context independent functions which provide functionality required by both the Adc and Adc2 modules. In order to operate in any given application, the function design must not write to any fixed static variable location, unless it is in Globally shared memory. All static variable writes outside of Globally shared memory must be performed via pointer access where the caller provides the pointer reference to allowed writable memory in the application context from with the caller is executing.

The motivation for creating core functions is to reduce duplication and testing of common code design.

# Figures

## Component Diagram

None

# Variable Data Dictionary

For details on module input / output variable, refer to the Data Dictionary for the application. Input / output variable names are listed here for reference.

|  |  |
| --- | --- |
| Module Inputs | Module Outputs |
| None | None |

## Module Internal Variables

This section identifies the name, range and resolutions for module specific data created by this module. If there are no range restrictions on the variable, the term “FULL” is placed into the table for legal range.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| None |  |  |  |  |

### User defined typedef definition/declaration

This section documents any user types uniquely used for the module.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| None |  |  |  |  |

# Constant Data Dictionary

## Calibration Constants

This section lists the calibrations used by the module. For details on calibration constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| None |

## Program(fixed) Constants

### Embedded Constants

All embedded constants whose values are provided in Eng units will be evaluated to the equivalent counts by using the FPM\_InitFixedPoint\_m() macro within the #define statement.

#### Local

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| D\_NUMOFADCCALREADS\_CNT\_U8 | 1 | Counts | 8 |

#### Global

This section lists the global constants used by the module. For details on global constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| D\_FALSE\_CNT\_LGC |
| D\_TRUE\_CNT\_LGC |

### Module specific Lookup Tables Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Value | Software Segment |
| None |  |  |  |

# Functions/Macros used by the Sub-Modules

## Library Functions / Macros

The library and functions / Macros that are called by the various sub modules are identified below,

1. None

## Data Hiding Functions

1. <None>

## Global Functions/Macros Defined by this Module

### Offset Calibration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | ADCOffsetCalibration | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | adcREG | adcctrlregs\_t\* | FULL | FULL |  |
| **Return Value** | Status\_Cnt\_T\_u08 | uint8 | FULL | FULL |  |

Perform the ADC Calibration and Storing of offset.

#### Calibration Offset



## Local Functions/Macros Used by this MDD only

# Software Module Implementation

## Runtime Environment (RTE) Initial Values

None

## Initialization Functions

None

## Periodic Functions

None

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

None

## Transition Functions

None

# Execution Requirements

## Execution Sequence of the Module

No integration scheduling required. The only service offered by this module has a fixed scheduling within the Adc subsystem.

## Execution Rates for sub-modules called by the Scheduler

This table serves as reference for the Scheduler design

|  |  |  |
| --- | --- | --- |
| Function Name | Calling Frequency | System State(s) in which the function is called |
| None |  |  |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| N/A |  |

# Memory Map Definition Requirements

## Sub Modules (Functions)

This table identifies the software segments for functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| ADCOffsetCalibration | ADC\_START\_SEC\_CODE |

## Local Functions

This table identifies the software segments for local functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
|  |  |

# Known Issues / Limitations With Design

1. INLINE functions defined in “GlobalMacro.h” are not unit tested

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial version | 24Apr13 | Selva |