**Motor Control Manager**

**FDD #AR300A**

.

[1. High Level Description 4](#_Toc441146862)

[2. Derived Requirements 4](#_Toc441146863)

[3. Function I/O 5](#_Toc441146864)

[3.1. Data Ownership 5](#_Toc441146865)

[3.2. Input Description 5](#_Toc441146866)

[3.3. Output Description 6](#_Toc441146867)

[3.4. Sub-Function Data Flow 7](#_Toc441146868)

[4. Design Rationale and Assumptions 8](#_Toc441146869)

[5. Sub-Functions 8](#_Toc441146870)

[5.1. Sub-Function: MotCtrlMgrPer1 – Motor Control To 2ms RTE interface 8](#_Toc441146871)

[5.1.1. Hardware Related Design 8](#_Toc441146872)

[5.1.2. Software Related Design 9](#_Toc441146873)

[5.1.3. Sub Function Calibrations 9](#_Toc441146874)

[5.1.4. Signal Availability 9](#_Toc441146875)

[5.2. Sub-Function: MotCtrlMgrPer2 –2ms RTE to Motor Control interface 9](#_Toc441146876)

[5.2.1. Hardware Related Design 9](#_Toc441146877)

[5.2.2. Software Related Design 9](#_Toc441146878)

[5.2.3. Sub Function Calibrations 10](#_Toc441146879)

[5.2.4. Signal Availability 10](#_Toc441146880)

[5.3. Sub-Function: MotCtrlMgrIrq: Motor Control Interrupt Service Routine 10](#_Toc441146881)

[5.3.1. Hardware Related Design 10](#_Toc441146882)

[5.3.2. Software Related Design 10](#_Toc441146883)

[5.3.3. Sub Function Calibrations 10](#_Toc441146884)

[5.3.4. Signal Availability 10](#_Toc441146885)

[5.4. Sub-Function: Definition of Motor Control Data 10](#_Toc441146886)

[5.4.1. Hardware Related Design 10](#_Toc441146887)

[5.4.2. Software Related Design 10](#_Toc441146888)

[5.4.3. Sub Function Calibrations 11](#_Toc441146889)

[5.4.4. Signal Availability 11](#_Toc441146890)

[5.5. Sub-Function: Non-RTE Enumeration Definitions 12](#_Toc441146891)

[5.5.1. Hardware Related Design 12](#_Toc441146892)

[5.5.2. Software Related Design 12](#_Toc441146893)

[5.5.3. Sub Function Calibrations 12](#_Toc441146894)

[5.5.4. Signal Availability 12](#_Toc441146895)

[5.6. Sub-Function: Motor Control Data Access Macros 12](#_Toc441146896)

[5.6.1. Hardware Related Design 12](#_Toc441146897)

[5.6.2. Software Related Design 12](#_Toc441146898)

[5.6.3. Sub Function Calibrations 12](#_Toc441146899)

[5.6.4. Signal Availability 13](#_Toc441146900)

[5.7. Sub-Function: Motor Control Data Signal Mapping 13](#_Toc441146901)

[5.7.1. Hardware Related Design 13](#_Toc441146902)

[5.7.2. Software Related Design 13](#_Toc441146903)

[5.7.3. Sub Function Calibrations 13](#_Toc441146904)

[5.7.4. Signal Availability 13](#_Toc441146905)

[6. Timing / Execution Constraints 13](#_Toc441146906)

[6.1. Rationale / Comments 13](#_Toc441146907)

[6.2. Rates and State Execution 14](#_Toc441146908)

[7. Serial Communications Interfaces 14](#_Toc441146909)

[8. Additional Information 14](#_Toc441146910)

[9. Revision Record & Change Approval 15](#_Toc441146911)

# High Level Description

The Motor Control Manager component is responsible for three major tasks:

1. Defining and owning the Motor Control Interrupt routine
2. Defining and owning all Motor Control related global signals that are not handled by the RTE including providing interfaces for component access to these signals
3. Providing a standardized interface between the Motor Control related global signals and the RTE signals

# Derived Requirements

None

# Function I/O



## Data Ownership

The following table shows the data that the MotCtrlMgr is expected to define and own. Please note that only a subset of this data is explicitly used as I/O for the MotCtrlMgr subfunctions.

|  |  |
| --- | --- |
| Data | Description |
| MotCtrlMgr\_MotCtrlToTwoMilliSec\_Rec | Structure containing all of the signals that are written by motor control scheduled runnables that are required to be read by 2ms RTE scheduled runnables. This structure is the structure that the Motor Control Runnables write to. The list of signals contained in this structure can change from program to program based on program dataflow requirements. |
| MotCtrlMgr\_TwoMilliSecFromMotCtrl\_Rec | Structure containing all of the signals that are written by motor control scheduled runnables that are required to be read by 2ms RTE scheduled runnables. This structure is the structure that the 2ms RTE scheduled runnables read from. The list of signals contained in this structure can change from program to program based on program dataflow requirements. |
| MotCtrlMgr\_TwoMilliSecToMotCtrl\_Rec | Structure containing all of the signals that are written by 2ms RTE scheduled runnables that are required to be read by motor control scheduled runnables. This structure is the structure that the 2ms RTE scheduled runnables write to. The list of signals contained in this structure can change from program to program based on program dataflow requirements. |
| MotCtrlMgr\_MotCtrlFromTwoMilliSec\_Rec | Structure containing all of the signals that are written by 2ms RTE scheduled runnables that are required to be read by motor control scheduled runnables. This structure is the structure that the Motor Control Runnables read from. The list of signals contained in this structure can change from program to program based on program dataflow requirements. |
| MotCtrlMgr\_MotCtrlInt\_Rec | Structure containing all of the signals that read and written only by motor control scheduled runnables (i.e. no interface with RTE scheduled runnables required). The list of signals contained in this structure can change from program to program based on program dataflow requirements. |

## Input Description

The following inputs are used by this FDD. Source FDD, range, resolution are located in the FDD data dictionary.

|  |  |
| --- | --- |
| Input Name | Description |
| MotCtrlMgr\_TwoMilliSecFromMotCtrl\_Rec | See description above |
| *<Signal1>…<Signal#>* | All RTE signals coming from the 2ms RTE scheduled runnables that are required to be read by motor control scheduled runnables. The list of signals can change from program to program based on program dataflow requirements. |

## Output Description

The following outputs are generated by this FDD. Source FDD, range, resolution are located in the FDD data dictionary.

|  |  |
| --- | --- |
| Output Name | Description |
| MotCtrlMgr\_TwoMilliSecToMotCtrl\_Rec | See description above |
| *<SignalA>…<SignalX>* | All RTE signals being read by 2ms RTE scheduled runnables that are written by motor control scheduled runnables. The list of signals can change from program to program based on program dataflow requirements. |

## Sub-Function Data Flow



# Design Rationale and Assumptions

The Motor Control Manager component is intended to provide a flexible design that will allow a consistent definition of all of the Motor Control related global signals, the Motor Control interrupt routine, and the interface between the Motor Control global signals and the RTE. The design is also partially influenced on the knowledge of how the DMA, ADC, and TSG3 hardware peripherals will be used for electrical architecture 4. The following assumptions were considered in the design, and further details on some of these items can be found later in this document:

* The ADC0 will be used primarily for Motor Control ADC reads and the ADC1 will be used for 2ms ADC reads. The 2ms ADC reads will require explicit transfer from the Motor Control time domain to the 2ms time domain.
* The DMA will be transferring ADC results from ADC result registers into RAM and therefore the ADC result RAM signals must be packed together in proper order. Additionally, all ADC results registers will be transferred, not just those that are used in a program.
* The DMA will be transferring TSG3 PWM signals from RAM to TSG3 registers and therefore must be packed together in proper order. The signals names and ordering are as follows:
  + MotCtrlTSG3*n*DCMP0E, MotCtrlTSG3*n*CMP0E, MotCtrlTSG3*n*CMP12E
  + MotCtrlTSG3*n*CMPWE, MotCtrlTSG3*n*CMPVE, MotCtrlTSG3*n*CMPUE
* All signals being transferred between the Motor Control and 2ms time domains will be done through the DMA and will require definition of signals for both time domains. This is why these signals are all packed together in a structure. Additionally, it is assumed that the DMA will be using 128 bit transfers for this data, and memory alignment and size of the structure are designed accordingly.
* It is assumed that the RAM data structures that the DMA will be writing to will need to be placed in an isolated RAM location in memory and will therefore need special provisions to allow specific placement of this data in memory. It is also assumed the DMA will not have any specific restrictions on memory that it can read from.
* Signals owned by the Motor Control Manager are not required to be defined as a structure. Enumerations, however, will be supported, but only for underlying datatypes of uint8, uint16, uint32.
* The current design assumes only a 2ms interface to RTE is required

# Sub-Functions

## Sub-Function: MotCtrlMgrPer1 – Motor Control To 2ms RTE interface

### Hardware Related Design

None

### Software Related Design



### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: MotCtrlMgrPer2 –2ms RTE to Motor Control interface

### Hardware Related Design

None

### Software Related Design



### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: MotCtrlMgrIrq: Motor Control Interrupt Service Routine

### Hardware Related Design

None

### Software Related Design

The motor control interrupt routine needs to be configurable to allow flexibility in the runnable list that it contains as well as the order in which the runnables are called. Additionally, it will contain a loop counter that will toggle between 0 and 1 every motor control loop. Any runnables that need to be scheduled at a rate of MotorControlx2 will only run when the counter is equal to 1, whereas runnables that need to be scheduled at a rate of MotorControl will run regardless of the counter value. The counter should be initialized to 1 and will toggle after all runnables are executed.

### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: Definition of Motor Control Data

### Hardware Related Design

None

### Software Related Design

This sub-function defines the details of how the process for defining the motor control related signals contained in the Data Ownership section above. There are some high level details that hold true for all signals as follows:

* Structures are used to implement the different data categories to group all of the data together
* All structure elements are defined to pack the data into a structure in the most efficient manner. This involves grouping all of the same size datatypes together in the structure as well as grouping the different size groups in descending order of size (i.e. 32bit->16bit->8bit).
* All structures are aligned to start at an address that is evenly divisible by 16 to allow for 128bit DMA transfers. Additionally, all structures will contain some pad bytes at the end of the structure to make the structure size (in bytes) evenly divisible by 16.
* The following properties are needed to define the structure details:
  + Signal Name (without “MotCtrl” prefix if applicable)
  + Underlying signal datatype (float32, uint32, sint32, uint16, sint16, uint8, sint8, boolean)
    - If signal is enumerated the enumeration name is also needed
  + Initial Value of each signal
  + Size (for array handling)
  + Signal usage for each Motor Control related signal:
    - Signal read in a 2ms (or higher) function
    - Signal written in a 2ms (or higher) function
    - Signal read in Motor Control function
    - Signal written in a Motor Control function

#### MotCtrlMgr\_MotCtrlToTwoMilliSec\_Rec

If the motor control signal is written in a Motor Control function and read in a 2ms function, it would need to be placed in this structure. All signals in this structure should be prepended with “MotCtrl” given that this structure is accessed by Motor Control functions.

This structure should contain ADC1 result data as well given that it needs to be read by 2ms functions. This result data needs to be aligned at an address that is evenly divisible by 16 to allow for 128bit DMA transfers.

#### MotCtrlMgr\_TwoMilliSecFromMotCtrl\_Rec

This structure is an exact duplicate of MotCtrlMgr\_MotCtrlToTwoMilliSec\_Rec except “MotCtrl” is not prepended to each signal because this structure is accessed by 2ms functions.

#### MotCtrlMgr\_TwoMilliSecToMotCtrl\_Rec

If the motor control signal is written in a 2ms function and read in a Motor Control function, it would need to be placed in this structure.

#### MotCtrlMgr\_MotCtrlFromTwoMilliSec\_Rec

This structure is an exact duplicate of MotCtrlMgr\_TwoMilliSecToMotCtrl\_Rec except “MotCtrl” is prepended to each signal because this structure is accessed by Motor Control functions.

#### MotCtrlMgr\_MotCtrlInt\_Rec

If the motor control signal is written in a Motor Control function and is not read in a 2ms function, it would need to be placed in this structure.

This structure should contain ADC0 result data as well given that it needs to be read by Motor Control functions. This result data needs to be aligned at an address that is evenly divisible by 16 to allow for 128bit DMA transfers.

This structure should also contain TSG3 data as described in the above Design Assumptions section. This data must also be ensured to be aligned at an address that is evenly divisible by 4 to allow 32bit DMA transfers.

### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: Non-RTE Enumeration Definitions

### Hardware Related Design

None

### Software Related Design

Since certain signals that reside outside of the RTE may be required to be defined as enumerations, it is possible that there will not be visibility to the RTE defined enumerations. To resolve this, the Motor Control Manager will also allow definition of enumeration types and enumeration element names and values. This will need to be configurable on an as-needed basis. The following information is needed for this definition:

* Enumeration Name
* Underlying Enumeration Datatype (assumed uint8, uin16, and uint32 possibilities)
* Enumeration Element Names and Values

### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: Motor Control Data Access Macros

### Hardware Related Design

None

### Software Related Design

To abstract the underlying structure definition for each Motor Control signal, the Motor Control Manager will define access macros for external software modules to use when accessing from Motor Control functions. These will either be accessing elements of MotCtrlMgr\_MotCtrlInt\_Rec, MotCtrlMgr\_MotCtrlToTwoMilliSec\_Rec, or MotCtrlMgr\_MotCtrlFromTwoMilliSec\_Rec depending on the signal access needs. The macros will be defined with the format:

MOTCTRLMGR\_<*SignalIdentifier*>

Please note <*SignalIdentifier*> should always include the “MotCtrl” prefix since all signals accessed in Motor Control functions require this.

### Sub Function Calibrations

None

### Signal Availability

None

## Sub-Function: Motor Control Data Signal Mapping

### Hardware Related Design

None

### Software Related Design

In order to resolve the fact that names for inputs and outputs will not always align, the motor control manager must take into account this when creating the Motor Control Data Access Macros (see corresponding Sub-Function in this document). The strategy adopted is that the access macros for inputs that connect to outputs of differing names will “map” to the data access macros for the outputs.

For example:

Input Name: *MotCtrlSignalA*

Output Name: *MotCtrlSignal1*

MOTCTRLMGR\_<*MotCtrlSignalA*> access macro would “map” to MOTCTRLMGR\_<*MotCtrlSignal1*> access macro and therefore ultimately be accessing the underlying structure element for *MotCtrlSignal1.* There would not be a separate structure element for *MotCtrlSignalA.*

Please note that for any given output, there could be multiple corresponding input signal names that it could “map” to.

### Sub Function Calibrations

None

### Signal Availability

None

# Timing / Execution Constraints

## Rationale / Comments

MotCtrlMgrPer1 should execute near the start of the 2ms loop before any other functions that need data that is output by runnables running at the Motor Control rate. The DMA transfer that is populating MotCtrlMgr\_TwoMilliSecFromMotCtrl\_Rec shall run before MotCtrlMgrPer1 and ideally enough other functionality runs in-between these two events that the transfer can complete with minimal wait time in MotCtrlMgrPer1 (since this function is waiting for DMA transfer completion).

MotCtrlMgrPer2 should execute as the final function of the forward path to allow minimal lag on computation of new PWM commands in the Motor Control loop.

## Rates and State Execution

|  |  |
| --- | --- |
| **Sub-Function Name** | **Rate (ms)** |
| MotCtrlMgrPer1 | 2 (all system states) |
| MotCtrlMgrPer2 | 2 (all system states) |

# Serial Communications Interfaces

None

# Additional Information

# Revision Record & Change Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Change Control #** | **Change Description** |
| 1 | 04/22/15 | EA4#510 | Initial Version |
| 2 | 01/21/16 | EA4#3424 | Added signal mapping subfunction |
|  |  |  |  |