# Module -- I2C Nexteer

This document describes the software design and implementation of the Nexteer inter-integrated circuit (I2C) driver for EA3.x applications.

# Equations for Register Settings

Depending on the type of device attached to the I2C bus different register settings may be required. The following equations were used to determine the module clock frequency and high and low times for the required device that are documented in this MDD.

## Module Clock Frequency

The module clock frequency determines the frequency at which the I2C module operates. The value in the prescale register (I2CPSC) is programmable and divides the input clock to produce the module clock. The module clock frequency must be between 6.7 MHz and 13.3 MHz for proper operation of the I2C module. At the time this specification was created, the input clock frequency is 80MHz.

## Master Clock Frequency

The master clock frequency is the frequency that will be used on the SCL pin of the TMS570 device when in master mode. Depending on the value of the I2CPSC and the desired clock low (I2CCKL) and high (I2CCKH) times, the mast clock frequency can be calculated by one of the two equations below:

Where *d* depends on:

|  |  |
| --- | --- |
| I2CPSC | d |
| 0 | 7 |
| 1 | 6 |
| Greater than 1 | 5 |

# 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 |
| I2cNxtr\_I2cTransfer\_Cnt\_M\_str. Mode\_Cnt\_b32 | 1 | 0 | 0x10 | I2CNXTR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| I2cNxtr\_I2cTransfer\_Cnt\_M\_str. Length\_Cnt\_u32 | 1 | FULL | FULL | I2CNXTR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| I2cNxtr\_I2cTransfer\_Cnt\_M\_str. DataPtr\_Cnt\_u08 | 1 | FULL | FULL | I2CNXTR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |

### 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) |
| i2cctrlregs\_t | OAR  IMR  STR  CLKL  CLKH  CNT  DRR  SAR  DXR  MDR  IVR  EMDR  PSC  PID11  PID12  DMAC  FUN  DIR  DIN  DOUT  SET  CLR  ODR  PD  PSL | uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32  uint32 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | 1023  255  32767  65535  65535  65535  255  1023  255  65535  4095  3  255  65535  255  3  1  3  3  3  3  3  3  3  3 |
| I2cTransferType | Mode\_Cnt\_b32  Length\_Cnt\_u32  DataPtr\_Cnt\_u08 | uint32  uint32  \*uint8 | FULL  FULL  FULL | FULL  FULL  FULL |
| enum i2cMode | I2C\_FD\_FORMAT  I2C\_START\_BYTE  I2C\_RESET\_OUT  I2C\_DLOOPBACK  I2C\_REPEATMODE  I2C\_10BIT\_AMODE  I2C\_TRANSMITTER  I2C\_MASTER  I2C\_STOP\_COND  I2C\_START\_COND  I2C\_FREE\_RUN  I2C\_NACK\_MODE | uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16 | 0x0008  0x0010  0x0020  0x0040  0x0080  0x0100  0x0200  0x0400  0x0800  0x2000  0x4000  0x8000 | 0x0008  0x0010  0x0020  0x0040  0x0080  0x0100  0x0200  0x0400  0x0800  0x2000  0x4000  0x8000 |
| enum i2cBitCount | I2C\_2\_BIT  I2C\_3\_BIT  I2C\_4\_BIT  I2C\_5\_BIT  I2C\_6\_BIT  I2C\_7\_BIT  I2C\_8\_BIT | uint16  uint16  uint16  uint16  uint16  uint16  uint16 | 0x2  0x3  0x4  0x5  0x6  0x7  0x0 | 0x2  0x3  0x4  0x5  0x6  0x7  0x0 |
| enum i2cIntFlags | I2C\_AL\_INT  I2C\_NACK\_INT  I2C\_ARDY\_INT  I2C\_RX\_INT  I2C\_TX\_INT  I2C\_SCD\_INT  I2C\_AAS\_INT  I2C\_RECV\_OVERRUN | uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16 | 0x0001  0x0002  0x0004  0x0008  0x0010  0x0020  0x0040  0x0800 | 0x0001  0x0002  0x0004  0x0008  0x0010  0x0020  0x0040  0x0800 |
| enum i2cStatFlags | I2C\_AL  I2C\_NACK  I2C\_ARDY  I2C\_RXRDY  I2C\_TXRDY  I2C\_SCD  I2C\_ADDRZEROST  I2C\_ADDRASSLV  I2C\_XSMT  I2C\_RXFULL  I2C\_BUSBUSY  I2C\_NACKMOD  I2C\_SLAVEDIR | uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16  uint16 | 0x0001  0x0002  0x0004  0x0008  0x0010  0x0020  0x0100  0x0200  0x0400  0x0800  0x1000  0x2000  0x4000 | 0x0001  0x0002  0x0004  0x0008  0x0010  0x0020  0x0100  0x0200  0x0400  0x0800  0x1000  0x2000  0x4000 |
| enum i2cDMA | I2C\_TXDMA  I2C\_RXDMA | uint16  uint16 | 0x20  0x10 | 0x20  0x10 |
| struct g\_i2cTransfer | mode  length  \*data | uint32  uint32  uint8 | Full  Full  N/A | Full  Full  N/A |

# 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\_CLKRESETPULSES\_CNT\_U08 | 1 | Counts | 18 |
| D\_CLOCKPERIOD\_US\_U08 | 1 | uS | 5 |

#### 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 |
| None |

### Module specific Lookup Tables Constants

(This is for lookup tables (arrays) with fixed values, same name as other tables)

|  |  |  |  |
| --- | --- | --- | --- |
| 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

### I2C Init

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_Init | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Enable

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_Enable | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Reset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_Reset | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Setup Master Transmit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetupMasterTransmit | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | SlaveAddress\_Cnt\_T\_u08 | uint8 | In | 0 | 127 |  |
|  | DataLength\_Cnt\_T\_u16 | uint16 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Setup Master Receive

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetupMasterReceive | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | SlaveAddress\_Cnt\_T\_u08 | uint8 | In | 0 | 127 |  |
|  | DataLength\_Cnt\_T\_u16 | uint16 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Switch Master Receive

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SwitchMasterReceive | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | DataLength\_Cnt\_T\_u16 | uint16 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Count

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetCount | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Count\_Cnt\_T\_u16 | uint16 | In | 0 | 65535 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Own Address

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetOwnAdd | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Address\_Cnt\_T\_u16 | uint16 | In | 1 | 127 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Slave Address

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetSlaveAdd | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Address\_Cnt\_T\_u16 | uint16 | In | 1 | 127 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Functional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetFunctional | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Port\_Cnt\_T\_u08 | Uint8 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Baudrate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetBaudrate | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Baud\_Hz\_T\_u32 | uint32 | In | 1 | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Is TX Ready

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_IsTxReady | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Ready\_Cnt\_T\_lgc | boolean | Out | FALSE | TRUE | N/A |

#### Description



### I2C Send Byte

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SendByte | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Byte\_Cnt\_T\_u08 | uint8 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Send

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_Send | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Length\_Cnt\_T\_u32 | uint32 | In | FULL | FULL |  |
|  | DataPtr\_Cnt\_T\_u08 | \*uint8 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Is RX Ready

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_IsRxReady | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Ready\_Cnt\_T\_lgc | boolean | Out | FALSE | TRUE |  |

#### Description



### I2C RX Error

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_RxError | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Status\_Cnt\_T\_b32 | uint32 | Out | FULL | FULL |  |

#### Description



### I2C Receive Ready

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_ReceiveByte | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Data\_Cnt\_T\_u08 | uint8 | Out | FULL | FULL |  |

#### Description



### I2C Set Receive

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetRecv | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Length\_Cnt\_T\_u32 | uint32 | In | FULL | FULL |  |
|  | DataPtr\_Cnt\_T\_u08 | \*uint8 | Out | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Direction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetDirection | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Dir\_Cnt\_T\_u08 | uint8 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Set Bit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetBit | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Bit\_Cnt\_T\_u08 | uint8 | In | 0 | 31 |  |
|  | Value\_Cnt\_T\_u08 | uint8 | In | 0 | 1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Get Bit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_GetBit | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Bit\_Cnt\_T\_u08 | uint8 | In | 0 | 31 |  |
| **Return Value** | Value\_Cnt\_T\_u08 | uint8 | Out | 0 | 1 |  |

#### Description



### I2C Enable Notification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_EnableNotification | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Flags\_Cnt\_T\_b32 | uint32 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Disable Notification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_DisableNotification | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Flags\_Cnt\_T\_b32 | uint32 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Generate Start Condition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_GenStartCond | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Generate Stop Condition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_GenStopCond | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



### I2C Get Interrupt Vector

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_GetIntVect | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Vector\_Cnt\_T\_u08 | uint8 | Out | 0 | 7 |  |

#### Description



### I2C Get Status

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_GetStatus | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
| **Return Value** | Status\_Cnt\_T\_u16 | uint16 | Out | FULL | FULL |  |

#### Description



### I2C Set Status

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | I2c\_SetStatus | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | I2cRegPtr\_Cnt\_T\_str | i2cctrlregs\_t | In/Out | See section 2.1.1 | See section 2.1.1 |  |
|  | Status\_Cnt\_T\_u16 | uint16 | In | FULL | FULL |  |
| **Return Value** | N/A |  |  |  |  |  |

#### Description



## Local Functions/Macros Used by this MDD only

None

# Software Module Implementation

## Runtime Environment (RTE) Initial Values

This section lists the initial values of data written by this module but controlled by the RTE. After RTE initialization, the data in this table will contain these values.

|  |  |
| --- | --- |
| Data | Value |
| None |  |

## Initialization Functions

None

## Periodic Functions

None

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

### Isr: Isr\_I2c

#### Design Rationale

None

#### Program Flow Start

Metrics\_TaskStart(D\_I2CNXT\_CNT\_U08)

#### I2c Notification



#### Program Flow End

Metrics\_TaskEnd(D\_I2CNXT\_CNT\_U08)

## Serial Communication Functions

None

# Execution Requirements

## Execution Sequence of the Module

(Describe in words relevant details about the execution sequence of the different sub modules.)

## 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) |
| None |  |

# 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 |
| I2c\_Init | N/A |
| I2c\_Enable | N/A |
| I2c\_Reset | N/A |
| I2c\_SetupMasterTransmit | N/A |
| I2c\_SetupMasterReceive | N/A |
| I2c\_SwitchMasterReceive | N/A |
| I2c\_SetCount | N/A |
| I2c\_SetOwnAdd | N/A |
| I2c\_SetSlaveAdd | N/A |
| I2c\_SetFunctional | N/A |
| I2c\_SetBaudrate | N/A |
| I2c\_IsTxReady | N/A |
| I2c\_SendByte | N/A |
| I2c\_Send | N/A |
| I2c\_IsRxReady | N/A |
| I2c\_RxError | N/A |
| I2c\_ReceiveByte | N/A |
| I2c\_SetRecv | N/A |
| I2c\_SetDirection | N/A |
| I2c\_SetBit | N/A |
| I2c\_GetBit | N/A |
| I2c\_EnableNotification | N/A |
| I2c\_DisableNotification | N/A |
| I2c\_GenStartCond | N/A |
| I2c\_GenStopCond | N/A |
| I2c\_GetIntVect | N/A |
| I2c\_GetStatus | N/A |
| I2c\_SetStatus | N/A |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| None |  |

# Known Issues / Limitations With Design

1. None

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1 | Initial component creation | 22-Aug-13 | Jared |
| 2 | 2 | Add Metrics hook. | 7-Oct-13 | BWL |
| 3 | 3 | New enum is created and receiver overrun is checked when receive data | 24-Feb-14 | Rijvi |
| 4 | 4 | Updated init code to add new bit-banged SCK strobe for A6836 | 3-Jun-14 | Jared |