# Module --

# High-Level Description

This module is responsible for the transport of sensor data between the I2C peripheral and the DigColPs Component. This module accomplishes this task by providing a number of interface functions. The functions handle the complexities of making request and handling the underlying interrupts to present the periodic task in the DigColPs module with the requested data on the next iteration.

# Figures

## Diagram – Function Data Sharing

See DigColPs\_MDD for sequence diagram and interaction between this module, the DigColPs module and the physical sensor.

# 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 | |
| Type\_Cnt\_u08 | | CommFaults\_Cnt\_b08 |
|  | | DataType\_Cnt\_u08 |
|  | | ColSnsrData\_Cnt\_u16 |
|  | | SpurSnsrData\_Cnt\_u16 |
|  | | DigColPsInt\_I2CHwCustData\_Uls\_M\_u16 |

## 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 |
| DigColPsInt\_CurrentStepNo\_Cnt\_M\_enum | 1 | 0 | 36 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DigColPsInt\_InitialTime\_mS\_M\_u32 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigColPsInt\_ColSnsrData\_Cnt\_M\_u16 | 1 | 0 | 65535 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_SpurSnsrData\_Cnt\_M\_u16 | 1 | 0 | 65535 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_ColSnsrErrData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_ColSnsrExtErrData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_ColSnsrCheckStatData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_SpurSnsrErrData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_SpurSnsrExtErrData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_SpurSnsrCheckStatData\_Cnt\_D\_u16 | 1 | 0 | FULL | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_CurrentSlave\_Cnt\_M\_u08 | 1 | 0 | 127 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_RecvdDataType\_Cnt\_M\_u08 | 1 | 0 | 5 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_Buffer\_Cnt\_M\_u08[] | 1 | 0 | 255 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_PrevReqDataType\_Cnt\_M\_u08 | 1 | 0 | 5 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_TransactionCnt\_Cnt\_M\_u08 | 1 | 0 | 255 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_PrevTransactionCnt\_Cnt\_M\_u0 8 | 1 | 0 | 255 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_InitFailedOnce\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_BusBusySeqError\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_NackOccured\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_SkipRegisterWrite\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_RecvOverrunError\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_SensInitialized\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_I2CHwCustData\_Uls\_M\_u16 | 1 | 0 | 511 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_I2CHwIncompleteCustData\_Uls\_M\_u16 | 1 | 0 | 255 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_16 |
| DigColPsInt\_AttempOccurForCustDatRead\_Cnt\_M\_u08 | 1 | 0 | 11 | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigColPsInt\_CmdFailOccurred\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_ColCustDatFound\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_SpurCustDatFound\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_ResetSensor\_Cnt\_M\_lgc | N/A | FALSE | TRUE | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| DigColPsInt\_IntrptSrc\_Cnt\_D\_u08 | N/A | 0U | 255U | DIGCOLPSINT\_START\_SEC\_VAR\_CLEARED\_8 |

### 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) |
| CommStepType | INIT\_NOT\_INITIALIZED  INIT\_SENSOR1\_READERROR\_SETREG  INIT\_SENSOR1\_READERROR\_READ  INIT\_SENSOR1\_READEXTERR\_SETREG  INIT\_SENSOR1\_READEXTERR\_READ  INIT\_SENSOR1\_SENDCMD  INIT\_SENSOR1\_CHECKSTAT\_SETREG  INIT\_SENSOR1\_CHECKSTAT\_READ  INIT\_SENSOR2\_READERROR\_SETREG  INIT\_SENSOR2\_READERROR\_READ  INIT\_SENSOR2\_READEXTERR\_SETREG  INIT\_SENSOR2\_READEXTERR\_READ  INIT\_SENSOR2\_SENDCMD  INIT\_SENSOR2\_CHECKSTAT\_SETREG  INIT\_SENSOR2\_CHECKSTAT\_READ  INIT\_SENSOR1\_EXTREADADDRREG\_SENDCMD  INIT\_SENSOR1\_EXTREADCTRLREG\_SENDCMD  INIT\_SENSOR1\_DUMMY\_SEND,  INIT\_SENSOR1\_DUMMY\_READ,  INIT\_SENSOR2\_EXTREADADDRREG\_SENDCMD  INIT\_SENSOR2\_EXTREADCTRLREG\_SENDCMD  INIT\_SENSOR2\_DUMMY\_SEND,  INIT\_SENSOR2\_DUMMY\_READ,  INIT\_SENSOR1\_EXTREADCTRLREG\_SETREG  INIT\_SENSOR1\_EXTREADCTRLREG\_READ  INIT\_SENSOR2\_EXTREADCTRLREG\_SETREG  INIT\_SENSOR2\_EXTREADCTRLREG\_READ  INIT\_SENSOR1\_EXTREADDATREG\_SETREG  INIT\_SENSOR1\_EXTREADDATREG\_READ  INIT\_SENSOR2\_EXTREADDATREG\_SETREG  INIT\_SENSOR2\_EXTREADDATREG\_READ  INIT\_COMPLETE  READ\_SENSOR1\_SETREG  READ\_SENSOR1\_GETDATA  READ\_SENSOR2\_SETREG  READ\_SENSOR2\_GETDATA  READ\_COMPLETE | uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8  uint8 | 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 |

# 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 |
| k\_ColSensorI2CAddress\_Cnt\_u08 |
| k\_I2CHWInitTransactionTime\_Sec\_f32 |
| k\_SpurSensorI2CAddress\_Cnt\_u08 |

## 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\_COLSENSOR\_CNT\_U08 | 1 | Cnt | 0 |
| D\_SPURSENSOR\_CNT\_U08 | 1 | Cnt | 1 |
| D\_CLEARSTATUSBITS\_CNT\_U16 | 1 | Cnt | 0x0007 |
| D\_RESETCMD\_CNT\_U16 | 1 | Cnt | 0x0746 |
| D\_EXTREADADDRREGCMD\_CNT\_U16 | 1 | Cnt | 0x0307 |
| D\_EXTREADCTRLREGCMD\_CNT\_U16 | 1 | Cnt | 0x8000 |
| D\_NACKOCCURRED\_CNT\_U08 | 1 | Cnt | 2 |
| D\_BBSEQERR\_CNT\_U08 | 1 | Cnt | 4 |
| D\_TRANSNOTCOMP\_CNT\_U08 | 1 | Cnt | 16 |
| D\_ANGLEDATA\_CNT\_U08 | 1 | Cnt | 1 |
| D\_NONE\_CNT\_U08 | 1 | Cnt | 0x00 |
| D\_ANGLEREG\_CNT\_U08 | 1 | Cnt | 0x20 |
| D\_CONTROLREG\_CNT\_U08 | 1 | Cnt | 0x1E |
| D\_ERRORREG\_CNT\_U08 | 1 | Cnt | 0x24 |
| D\_EXTERRORREG\_CNT\_U08 | 1 | Cnt | 0x26 |
| D\_STATUSREG\_CNT\_U08 | 1 | Cnt | 0x22 |
| D\_RECVOVERRUNERR\_CNT\_U08 | 1 | Cnt | 0x08 |
| D\_SENSINITDELAY\_MS\_U08 | 1 | mS | 32 |
| D\_CMDFAILED\_CNT\_U08 | 1 | Cnt | 0x20 |
| D\_EXTREADADDRREG\_CNT\_U08 | 1 | Cnt | 0x0A |
| D\_EXTREADCTRLREG\_CNT\_U08 | 1 | Cnt | 0x0C |
| D\_EXTREADDATREG\_CNT\_U08 | 1 | Cnt | 0x0E |
| D\_MAXATTEMPTSFORCUSTDATREAD\_CNT\_U08 | 1 | Cnt | 10 |
| D\_SENSINITNOTDONE\_CNT\_U08 | 1 | Cnt | 1<<7U (0x80) |
| D\_SECTOMILLSEC\_CNT\_F32 | 1 | Cnt | 1000.0F |
| D\_RESETDELAY\_MS\_U08 | 1 | mS | 7U |
| D\_NACKACKINTRPT1\_CNT\_U08 | 1 | Cnt | 1U |
| D\_NACKACKINTRPT2\_CNT\_U08 | 1 | Cnt | 2U |
| D\_ARBNLOSTINTRPT\_CNT\_U08 | 1 | Cnt | 4U |
| D\_RXROVRRUNERRINTRPT\_CNT\_U08 | 1 | Cnt | 8U |
| D\_STOPDTCTINTRPT\_CNT\_U08 | 1 | Cnt | 16U |
| D\_UKWNINTRPT\_CNT\_U08 | 1 | Cnt | 32U |

#### 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 |
| T\_DataRegisters\_Cnt\_u08[6] | 1 | D\_NONE\_CNT\_U08,  D\_ANGLEREG\_CNT\_U08,  D\_CONTROLREG\_CNT\_U08,  D\_ERRORREG\_CNT\_U08,  D\_EXTERRORREG\_CNT\_U08,  D\_STATUSREG\_CNT\_U08,  D\_EXTREADADDRREG\_CNT\_U08,  D\_EXTREADCTRLREG\_CNT\_U08,  D\_EXTREADDATREG\_CNT\_U08 | DIGCOLPSINT\_START\_SEC\_CONST\_8 |

# 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

### DigColPsInt\_Reset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_Reset | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | None |  |  |  |  |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



### Initialize

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_Init | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | None |  |  |  |  |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



### Get Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_GetData | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | ColSnsrDataPtr\_Cnt\_T\_u16 | uint16 | Out | FULL | FULL |  |
|  | SpurSnsrDataPtr\_Cnt\_T\_u16 | uint16 | Out | FULL | FULL |  |
|  | DataTypePtr\_Cnt\_T\_u08 | uint8 | Out | 0 | 5 |  |
| **Return Value** | CommFaults\_Cnt\_T\_b08 | uint8 | Out | 0 | 16 | 0 |

#### Description





### Start Request

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_StartRequest | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | Type\_Cnt\_T\_u08 | uint8 | Out | 0 | 5 |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



### Interrupt Notification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_InterruptNotification | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | Flags\_Cnt\_T\_b16 | uint16 | In | 1 | 64 |  |
| **Return Value** | None |  |  |  |  |  |

#### Design Rationale

The interrupt notification function needs to be setup by the integrator to be called by the Isr\_I2C interrupt service inside of the I2cNxtr component through the use of the I2cNxtr\_Cfg.h header file. A template for that header file can be found in the I2cNxtr includes directory.

In the FDD rev.005 the timeout requirement for ‘Customer Data’ read was 5 to 10 ms. Instead of timeout it is implemented by counting the ‘Number of Attempts’ occurred to read ‘Customer Data’, because it speeds up the interrupt execution time. ‘Number of Attempts’ required to fulfill the same timeout operation is determined in the following calculation.

Number of bits sends and receives by the master to determine whether the ‘Customer Data’ is ready to read is 47bits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **7** | **1** | **1** | **8** | **1** |
| **S bit** | **Slave addr.** | **W bit** | **ACK** | **CTRL reg addr.** | **ACK** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **7** | **1** | **1** | **8** | **1** | **8** | **1** |
| **S bit** | **Slave addr.** | **R bit** | **ACK** | **MSB of CTRL reg. data** | **ACK** | **LSB of CTRL reg. data** | **SP bit** |

**BLUE: From Master to Slave**

**GREEN: From Slave to Master**

So for 100Kb baud speed total time required to check control register data for both column and spur sensors is (47\*2)/100K = 0.94 ms

Therefore for 5 ms timeout ‘Number of Attempts’ occurred = 5/0.94 = 5.32

Similarly for 10 ms timeout ‘Number of Attempts’ occurred = 10/0.94 = 10.64

Number **10** is chosen in this implementation

#### Notification Processing

















### Get Cust Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | DigColPsInt\_GetCustData | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | None |  |  |  |  |  |
| **Return Value** | DigColPsInt\_I2CHwCustData\_Uls\_M\_u16 | Uint16 | Out | 0 | 511 | 0 |

#### Description



## Local Functions/Macros Used by this MDD only

### Setup Write - Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | SetupWriteData | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | Register\_Cnt\_T\_u08 | uint8 | In | 0 | 127 |  |
|  | Data\_Cnt\_T\_u16 | uint16 | In | FULL | FULL |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



### Setup Write - Register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | SetupWriteRegister | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | Register\_Cnt\_T\_u08 | uint8 | In | 0 | 127 |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



### Setup Read

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | SetupRead | Type | Dir. | Min | Max | UTP Tol. |
| **Arguments Passed** | None |  |  |  |  |  |
| **Return Value** | None |  |  |  |  |  |

#### Description



# 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

### None

## Serial Communication Functions

None

# Execution Requirements

## Execution Sequence of the Module

## 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 |
| DigColPsInt\_Init | N/A |
| DigColPsInt\_GetData | N/A |
| DigColPsInt\_StartRequest | N/A |
| DigColPsInt\_InterruptNotification | N/A |
| DigColPsInt\_GetCustData | 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. INLINE functions in GlobalMacro.h are not unit tested

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1 | Initial component creation | 21-Aug-13 | Jared |
| 2 | 2 | Implemented the FDD rev.003:  Changed inside the functions DigColPsInt\_Init(), DigColPsInt\_GetData(), DigColPsInt\_StartRequest() and DigColPsInt\_InterruptNotification().  Added delay before sensors initialization.  Added six new display variables.  Added receiver overrun error checking and action.  Added a bit for receive overrun error in CommFaults\_Cnt\_T\_b08 | 27-Feb-14 | Rijvi |
| 3 | 3 | Implemented the FDD rev.005.  Fixed anomaly 6089.  Fixed anomaly 6471. | 20-Mar-14 | Rijvi |
| 4 | 4 | Implemented the FDD rev.006 | 26-Mar-14 | Rijvi |
| 5 | 5 | Implemented the FDD rev 007 | 11-May-14 | Selva |
| 6 | 6 | Updated to FDD ver 008 | 27-Jun-14 | Jared |
| 7 | 7 | Updated to FDD ver12 compatibility with DSS | 05-Sep-14 | Vishnu |
| 8 | 8 | Unit Testing Finding Fixes | 13-Oct-14 | KPIT-PM |
| 9 | 9 | Updated to FDD ver 015 | 13-May-16 | JK |
| 10 | 10 | Updated to FDD ver 017 | 23-Jun-16 | JK |