

I2C™ Master Library Module (Polled)

1.	Introduction	2
2.	Module Features	2
3.	List of Component Modules.....	3
4.	Using the Library Module in a Project	3
5.	List of Shared Parameters	4
	<i>Shared Functions.....</i>	<i>4</i>
	<i>Shared Macros.....</i>	<i>4</i>
6.	Functions.....	5
7.	Macros.....	7
8.	Error and Status Flags	9

1. Introduction

The I2CMPol is a general-purpose library module. It configures MSSP module in Master mode and helps in communicating with the I2C™ Slave. If this library module is used with a device not having MSSP module, then the following message is displayed while compiling. "This controller does not have MSSP Module".

The module code is linkable and relocatable, which provides the user, the facility to use it without modifications.

By using this Module one can write his application to interact with any of the I2C Slaves like EEPROM, ADC, Digital Potentiometer, LCD etc.

The module allows user to concentrate more on his application's development by providing these library functions.

2. Module Features

It supports following features:-

- It provides simple and primitive functions to communicate with the I2C Slave.
- It generates Error flags on the occurrence of an error. All error conditions are passed through the 'W' Register.

3. List of Component Modules

I2CMPol.P16.ex.txt	This is an example file developed to demonstrate the use of the library functions for PIC16 family.
I2CMPol.P18.ex.txt	This is an example file developed to demonstrate the use of the library functions for PIC18 family.
I2CMPol.asm	This is the I2C Master code implementation file. <u>One needs to include this file in their project.</u>
16I2CMP.asm	This is the I2C Master code implementation file for PIC16 family. The I2CMPol.asm file will include this file if the PIC16 family processor is used.
18I2CMP.asm	This is the I2C Master code implementation file for PIC18 family. The I2CMPol.asm file will include this file if the PIC18 family processor is used.
I2CMPol.inc	This file contains the definitions of all the shared parameters and the macros. <u>One needs to include this in the Assembly file</u> where the library functions and macros are called. This file is taking care of definition of all Extern Global parameter so one can directly call library routines in their program.
P16xxx.inc	General purpose processor definition file for PIC16 family
P18xxx.inc	General purpose processor definition file for PIC18 family

4. Using the Library Module in a Project

Please follow the steps below to use this library module in your project.

1. Use the Application Maestro to configure the module as required.
2. At the 'Generate Files' step, save the output to the directory where your project code resides.
3. Launch MPLAB, and open the project's workspace.
4. Verify that the Microchip language tool suite is selected (*Project>Select Language Toolsuite*).
5. In the Workspace view, right-click on the "Source Files" node. Select the "Add Files" option. Select I2CMPol.asm and click **OK**.
6. Now right-click on the "Linker Scripts" node and select "Add Files". Add the appropriate linker file (.lkr) for the project's target microcontroller.
7. Add any other files that the project may require. Save and close the project.
8. In your main source (assembler) file, add include directive at the head of the code listing to include I2CMPol.inc. By doing so, all files required to make the generated code work in your project will be included by reference when you build the project.
9. To use the module in your application, invoke the functions or macros as needed.

5. List of Shared Parameters

Shared Functions

I2CMPolInit	It is used for Synchronous Serial Port Initialization It initializes Port according to the options opted through Application Maestro.
I2CMPolPut	It is used for transmitting a byte on I2C Bus.
I2CMPolGet	It is used for reading the received byte.
I2CMPolIsIdle	It is used for checking/waiting for Idle condition of I2C Bus.
I2CMPolIsDataReady	It is used for checking/waiting for reception of data on I2C Bus.
I2CMPolIsAckReceived	It is used for checking for reception of Acknowledge on I2C Bus.
I2CMPolIsBusCollision	It is used for checking for occurrence of I2C Bus Collision.

Shared Macros

mI2CMPolStart	Generates Start condition on I2C Bus.
mI2CMPolReStart	Generates Repeated Start condition on I2C Bus.
mI2CMPolStop	Generates Stop condition on I2C Bus.
mI2CMPolAck	Sends Acknowledgement on I2C Bus.
mI2CMPolNoAck	Sends No Acknowledgement on I2C Bus.
mI2CMPolEnableReceiver	Enables I2C receiver to receive a byte.
mI2CMPolDisable	Disables Synchronous Serial Port.

6. Functions

Function	I2CMPolInit
Preconditions	None
Overview	This function is used for initializing the MSSP module. It initializes the module according to Application Maestro options.
Input	Application Maestro options
Output	None
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep

Function	I2CMPolPut
Preconditions	'mI2CMPolStart' macro should have been invoked.
Overview	This function sends the byte in 'W' Reg. over I2C bus and checks for Write Collision.
Input	'W' Register.
Output	'W' Register. It will have: '0' - On proper initialization of transmission. 'I2CMErrWriteCollision' - On occurrence of the Write Collision error.
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep

Function	I2CMPolGet
Preconditions	'mI2CMPolEnableReceiver' macro should have been invoked and 'I2CMPolIsDataReady' should return a '0'.
Overview	This function reads the byte received.
Input	None
Output	'W' Register.
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep

Function	I2CMPolIsIdle
Preconditions	Must be called after every I2CMPol function and macro, except 'I2CMPolGet'
Overview	In Non Blocking Option – This function checks whether the I2C Bus is Idle. In Blocking Option – This function waits till the I2C Bus is Idle.
Input	None
Output	In Non Blocking Option – 'W' Register. It will have: '0' - If the I2C Bus is Idle. 'I2CMBusNotIdle' - If the I2C Bus is not Idle. In Blocking Option – None
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep

Function	I2CMPolIsDataReady
Preconditions	'mI2CMPolEnableReceiver' should have been invoked
Overview	In Non Blocking Option – This checks whether the Data is received. It also checks for the Over flow error. In Blocking Option – It checks for the Over flow error. If there is no error waits till Data is ready.
Input	None
Output	'W' Register. It will have: In Non Blocking Option – '0' - If the Data is ready 'I2CMDDataNotReady' - If Data is not ready 'I2CMErrRxDataOverFlow' - If Over flow error has occurred. In Blocking Option – 'I2CMErrRxDataOverFlow' - If Over flow error has occurred
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep
Function	I2CMPolIsAckReceived
Preconditions	'I2CMPolPut' should have been called
Overview	This checks whether acknowledge has been received.
Input	None
Output	'W' Register. '0' If Ack is received from Slave. 'I2CMErrRxNoAck' - If No Ack is received from Slave
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep
Function	I2CMPolIsBusCollision
Preconditions	Must be called after every I2CMPol function and macro, in Multi-Master mode. This portion of the code will be assembled only if opted for.
Overview	This checks whether the I2C Bus Collision has occurred.
Input	None
Output	'W' Register. '0' - If I2C Bus Collision has not occurred. 'I2CMErrBusCollision' - If I2C Bus Collision has occurred.
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep

7. Macros

Macro	mI2CMPolStart
Overview	Preconditions- 'I2CMPolInit' should have been called. This macro generates the Start condition on the I2C bus.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Macro	mI2CMPolReStart
Overview	Preconditions- At least once 'I2CMPolPut' should have been called. This macro generates Repeated Start condition on the I2C bus. This should be used if Start condition is to be generated, without generating a Stop condition for the previous Start condition.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Macro	mI2CMPolStop
Overview	Preconditions- 'mI2CMPolStart' should have been invoked. This macro generates the Stop condition on the I2C Bus.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Macro	mI2CMPolAck
Overview	Preconditions- 'I2CMPolGet' should have been called. This macro sends the Acknowledge on the I2C bus
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Macro	mI2CMPolNoAck
Overview	Preconditions- 'I2CMPolGet' should have been called. This macro sends the No Acknowledge on the I2C bus
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None

Macro	mI2CmpolEnableReceiver
Overview	Preconditions- 'I2CmpolIsAckReceived' should return no error, if it is invoked after calling 'I2CmpolPut' else the macro 'mI2CmpolAck' should have been invoked. It enables the receiver.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Macro	mI2CmpolDisable
Overview	Preconditions- 'mI2CmpolStop' should have been invoked. Disables the MSSP module.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None

8. Error and Status Flags

All errors/status are set as a content of 'W' Register. Individual errors/status are unique. Please refer below list for the information.

I2CMErrWriteCollision	This indicates that, Write collision has occurred while trying to transmit the byte.
I2CMErrNoAck	This indicates that, No Acknowledge is received from the Slave after transmitting the byte.
I2CMErrRxDataOverFlow	This indicates that, one more byte has been received before reading the previous byte.
I2CMErrBusCollision	This indicates that, the I2C Bus Collision has occurred. This can occur only in Multi-Master setup.
I2CMBusNotIdle	This indicates that, the I2C Bus is not yet Idle. This is to be checked only when Non Blocking option is opted.
I2CMDDataNotReady	This indicates that, the Data is not yet fully received. This is to be checked only when Non Blocking option is opted