Outlines

This application note is a reference material for developing products using the arbitration function in I2C interface (I2C) function of M3H Group (1). This document helps the user check operation of the product and develop its program.

Target sample program: I2C_EEPROM_arbitration
Table of Contents

Outlines ................................................................................................................................................. 1
Table of Contents .................................................................................................................................. 2
1. Preface .............................................................................................................................................. 3
2. Reference Document ........................................................................................................................ 4
3. Function to Use .................................................................................................................................. 4
4. Target Device ..................................................................................................................................... 4
5. Conditions for Correct Operation ...................................................................................................... 5
6. Evaluation Board Setting ................................................................................................................... 6
7. Operation of Evaluation Board ............................................................................................................ 7
8. Outline of I²C Interface function ........................................................................................................ 8
  8.1. Clock Supply ..................................................................................................................................... 8
9. Sample Program ................................................................................................................................... 9
  9.1. Initialization ....................................................................................................................................... 9
  9.2. Sample program main operation ...................................................................................................... 9
  9.3. Output Example of Sample Program ............................................................................................... 10
    9.3.1. Setting Example of Terminal Software ........................................................................................ 11
  9.4. Operating Flow of Sample Program ............................................................................................... 12
10. Precaution ....................................................................................................................................... 20
11. Revision History ............................................................................................................................. 20
RESTRICTIONS ON PRODUCT USE ................................................................................................ 21
1. Preface

This sample program is used to check the operation of the I2C communication function. First write data to I2C EEPROM. The function to read the written data via multiple CPUs is executed by the terminal software on the host PC via USB-UART.

Structure diagram of Sample program
2. Reference Document

- Datasheet
  TMPM3H group (1) datasheet Rev2.0 (Japanese edition)
- Reference manual
  I2C interface (I2C-B) Rev2.0 (Japanese edition)
- Other reference document
  TMPM3H(1) Group Peripheral Driver User Manual (Doxygen)

3. Function to Use

<table>
<thead>
<tr>
<th>IP</th>
<th>Channel</th>
<th>Port</th>
<th>Function / operation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2C interface</td>
<td>ch0</td>
<td>PC0 (I2C0SCL)</td>
<td>I2C mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PC1 (I2C0SDA)</td>
<td></td>
</tr>
<tr>
<td>Asynchronous communication</td>
<td>ch0</td>
<td>PA1 (UT0TXDA)</td>
<td>UART mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA2 (UT0RXD)</td>
<td></td>
</tr>
<tr>
<td>Input and Output ports</td>
<td></td>
<td>PN1 (Input port)</td>
<td>Input</td>
</tr>
</tbody>
</table>

4. Target Device

The target devices of this application note are as follows;

<table>
<thead>
<tr>
<th>Device 1</th>
<th>Device 2</th>
<th>Device 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMPM3H6FWFG</td>
<td>TMPM3H6FUFG</td>
<td>TMPM3H6FSFG</td>
</tr>
<tr>
<td>TMPM3H6FWDFG</td>
<td>TMPM3H6FUDFG</td>
<td>TMPM3H6FSDFG</td>
</tr>
<tr>
<td>TMPM3H5FWFG</td>
<td>TMPM3H5FUFG</td>
<td>TMPM3H5FSFG</td>
</tr>
<tr>
<td>TMPM3H5FWDFG</td>
<td>TMPM3H5FUDFG</td>
<td>TMPM3H5FSDFG</td>
</tr>
<tr>
<td>TMPM3H4FWUG</td>
<td>TMPM3H4FUUG</td>
<td>TMPM3H4FSUG</td>
</tr>
<tr>
<td>TMPM3H4FWFG</td>
<td>TMPM3H4FUFG</td>
<td>TMPM3H4FSFG</td>
</tr>
<tr>
<td>TMPM3H3FWUG</td>
<td>TMPM3H3FUUG</td>
<td>TMPM3H3FSUG</td>
</tr>
<tr>
<td>TMPM3H2FWUG</td>
<td>TMPM3H2FUUG</td>
<td>TMPM3H2FSUG</td>
</tr>
<tr>
<td>TMPM3H2FWUG</td>
<td>TMPM3H2FUUG</td>
<td>TMPM3H2FSUG</td>
</tr>
<tr>
<td>TMPM3H1FWUG</td>
<td>TMPM3H1FUUG</td>
<td>TMPM3H1FSUG</td>
</tr>
<tr>
<td>TMPM3H1FWUG</td>
<td>TMPM3H0FUDUG</td>
<td>TMPM3H0FMDUG</td>
</tr>
</tbody>
</table>

* This sample program operates on the evaluation board of TMPM3H6FWFG.
  If other function than the TMPM3H6 one is checked, it is necessary that CMSIS Core related files (startup file and I/O header file) should be changed properly.
  The BSP related file is dedicated to the evaluation board (TMPM3H6). If other function than the TMPM3H6 one is checked, the BSP related file should be changed properly.
5. Conditions for Correct Operation

- Used microcontroller: TMPM3H6FWFG
- Used board: TMPM3H6FWFG Evaluation Board (Product of Sensyst)
- Unified development environment: IAR Embedded Workbench for ARM 8.11.2.13606
- Unified development environment: μVision MDK Version 5.24.2.0
- Terminal software: Tera Term V4.96
- Sample program: V1100

Evaluation board (TMPM3H6FWFG Evaluation Board) (Top view)

For purchasing the board, refer to the following homepage. (http://www.chip1stop.com/)
6. Evaluation Board Setting
Two evaluation boards should be prepared. The following connections should be done.

Common connections for Board A and Board B

<table>
<thead>
<tr>
<th>CN5</th>
<th>Use</th>
<th>Through-hole No</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UART(RXD)</td>
<td>9-10</td>
<td>Connection</td>
</tr>
<tr>
<td></td>
<td>UART(TXD)</td>
<td>11-12</td>
<td>Connection</td>
</tr>
</tbody>
</table>

Board A

<table>
<thead>
<tr>
<th>CN9</th>
<th>Use</th>
<th>Through-hole No</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I2C(SCL)</td>
<td>1-2</td>
<td>Connection</td>
</tr>
<tr>
<td></td>
<td>I2C(SDA)</td>
<td>3-4</td>
<td>Connection</td>
</tr>
</tbody>
</table>

Board B

<table>
<thead>
<tr>
<th>CN9</th>
<th>Use</th>
<th>Through-hole No</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Push SW(S7)</td>
<td>25-26</td>
<td>Connection</td>
</tr>
</tbody>
</table>

Connections between the boards
Board A CN9 No2 and Board B CN9 No2 should be connected.
Board A CN9 No4 and Board B CN9 No4 should be connected.
Board A CN9 No26 and Board B CN9 No26 should be connected.
7. Operation of Evaluation Board

"/* #define BOARD_B */ " of "main.c" in the program which is written to Board B should be enabled.
The USB_UART pins of the boards and the PC should be connected with USB cables.
The PC executes the communication setting after start-up of the terminal software (Tera Term).
The reset button should be pushed down on Board B.
The communication starts according to the command which is input.
For the details of the operation, refer to “Sample Program Main Operation”.
8. Outline of I²C Interface function

The I²C can operate as a transceiver circuit of 1ch (SCL, SDA) in 1 unit circuit.

8.1. Clock Supply

When using I²C, please set a clock enabling bit corresponding with the fsys supply on/off register A ([CGFSYSENA]) or B ([CGFSYSENB]) and fc supply on/off register ([CGFCEN]) as “1” (clock supply). Please refer to "Clock Control and Operation Mode" of the reference manual for the details.
9. Sample Program

When SW (S4) is ON, an event procedure which uses the I2C function is executed. Then the following operations can be checked using the I2C interface; data read from the I2C EEPROM device, detection of generation of Bus busy error, and detection of generation of Bus arbitration error. Each event procedure can be checked by displaying its status on the terminal software.

9.1. Initialization

The following initialization is done after power is supplied. The port setting is executed after the initialization of each clock setting, the watchdog timer setting and the clock setting.

9.2. Sample program main operation

After the initialization, the “main” function is executed, and the following initialization is done.

1. BSP (Board Support Package) initialization
2. Application initialization
3. Initialization of the I2C interface for an EEPROM device.

After the above procedure, the following operations should be done on the terminal software (Tera Term) on the PC.

At first, when the switch (S7) on Board B is pushed down, the characters "toshibaABCDEFGHIJKLMNOPQRST" are stored to the EEPROM device on Board A through the I2C interface.

At the second, when you press switch (S7) on evaluation board B, it will try to read EEPROM data of evaluation board A via I2C and read from address 0 for evaluation board A and address 4 from evaluation board B.

One of the four following operations is done according to the timing to write the I2C interface after the push-down of the switch (S7) on Board B:

1. Board A: Read successful. Board B: Read successful.
2. Board A: Bus busy error. Board B: Read successful.

"command >" is displayed on Tera Term at start-up. Then the switch (S7) should be pushed down on Board B to write the data. After that, every push-down of the switch (S7) on Board B executes the read and display of the data.
9.3. Output Example of Sample Program

When the sample program operates, the command results are shown as follows;

**Board A**

```
command >
write data >
toshibaABCDEFGHIJKLMNOPQRST
read data >
toshibaABCDEFGHIJKLMNOPQRST
read data >
   bus busy error !!
read data >
toshibaABCDEFGHIJKLMNOPQRST
read data >
toshibaABCDEFGHIJKLMNOPQRST
```

**Board B**

```
command >
write data >
toshibaABCDEFGHIJKLMNOPQRST
read data >
ibaABCDEFGHIJKLMNOPQRST
read data >
ibaABCDEFGHIJKLMNOPQRST
read data >
   bus busy error !!
read data >
   arbitration error !!
```
9.3.1. Setting Example of Terminal Software

The operation of the terminal software (Tera Term) has been checked with the following settings.
9.4. Operating Flow of Sample Program

The operating flows of the sample program are shown in the following:
Creation and Initialization

**EEPROM_i2c_init()**

1. **bsp_i2c**
   - **i2c_init(address, sda, scl)**

2. **CG_FSYSENA**
   - **FSYSENA02 (PortC) Enable**
   - **FSYSENA20 (I2C) Enable**

3. **Port**: i2C_SCL and i2C_SDA settings

4. **Register assignment**
   - **i2c**

5. **i2c.reset(address)**
   - **I2C_reset(address)**
   - **I2C_frequency(100KHz)**
   - **I2C_get_clock_setting(address)**
   - **Frequency = I2C_get_clock_setting()**

6. **I2C initial value setting**
   - **I2C_init(address)**
   - **result = I2C_init(); Successful**

7. **result = EEPROM_i2c_init(); Successful**
"write" command procedure

result: %cmp
In the case of "write":

EEPROM_i2c_WritePage
(Transmission data Count of Transmission Bytes)

Generation of Transmission data
Byte0:Address High Byte
Byte1:Address Low Byte
Byte2 to n:Data

i2c_write
(Instance handle, Slave address, Transmission data,
Count of Transmission Bytes, stop = 1)

ref: i2c_check_bus_free

ref: i2c_start

ref: i2c_byte_write(Slave address)

loop[1..Count of Transmission Bytes]

ref: i2c_byte_write(Transmission data)

opt
[stop=1]

ref: i2c_stop

Count of Transmission Bytes = EEPROM_i2c_WritePage()

Terminal display
Write data is displayed.
"read" command procedure

EEPROM_i2c_ReadData
(Reception buffer, Byte count)

i2c_eeprom

bep_i2c

i2c

Generation of Transmission data
Byte0 : Address High Byte
Byte1 : Address Low Byte

i2c_write
(Instance handle, Slave address, Transmission data,
Count of Transmission Bytes, stop = 0)

Write count = i2c_write(-);

Count of Reception Bytes = EEPROM_i2c_ReadData(-);

In the case of "read":

Terminal display
Read data is displayed.

ref Refer to i2c_write in "write" command procedure
ref i2c_start
ref i2c_byte_write(Slave address)

ref loop[1,Count of Reception Bytes - 1]

ref i2c_byte_read(last=0)

ref i2c_byte_read(last=1)

opt[stop=1]

ref i2c_stop

Count of Reception Bytes = EEPROM_i2c_ReadData(-);
```plaintext
i2c_check_bus_free

result = :Successful

bus_free=1*
(Only update of the internal information)

i2c_start

result = i2c_start(:):Successful

Start=1*
(Only update of the internal information)
```
i2c_byte_write

(i2c_byte_write (Instance handle and Transmission data))

ACK result = i2c_byte_write(-):I2C_ACK

loop [I2C interrupt request is present.]

I2C_int_status(Instance handle)

I2C interrupt status=I2C_int_status(-):

I2C_get_ack(Instance handle)

ACK result =I2C_get_ack(-):ACK

I2C_write_data(Instance handle and Transmission data)

I2C_write_data(-):

I2C_start_condition(Instance handle and Slave address)

I2C_start_condition(-):

I2C_clear_int_status(Instance handle)

I2C_clear_int_status(-):

Bus free check

i2c_active(Instance handle)

i2c_active_data

Bus free check

i2c_master(Instance handle)

i2c_Master status

CR2 = (I2CxCR2_INIT | I2CxCR2_PIN_CLEAR);

bus busy error end

i2c_slave(Instance handle)

I2C_write_data(Instance handle and Transmission data)

I2C_write_data(-):

loop [start=0]

I2C_int_status(Instance handle)

I2C interrupt status=I2C_int_status(-):

I2C_get_ack(Instance handle)

ACK result =I2C_get_ack(-):ACK

ACK result =i2c_byte_write(-):I2C_ACK
- `i2c_byte_read(Instance handle)`
  - Reception data = `i2c_byte_read(-)`

- `I2C_clear_int_status(Instance handle)`
  - `I2C_clear_int_status(-)`

- `I2C_write_data(Instance handle, Dummy data)`
  - `I2C_write_data(-)`

- `I2C_int_status(Instance handle)`
  - `I2C_interrupt_status = I2C_int_status(-)`

- `I2C_read_data(Instance handle)`
  - Reception data = `I2C_read_data(-)`

Loop

- `I2C_interrupt_request is present.`
10. Precaution

When using the sample program with CPU other than TMPM3H6, please check operation sufficiently.

11. Revision History

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2018-03-29</td>
<td>-</td>
<td>First release</td>
</tr>
</tbody>
</table>
RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as "TOSHIBA".
Hardware, software and systems described in this document are collectively referred to as "Product".

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.

PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE"). Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combusions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT. For details, please contact your TOSHIBA sales representative.

- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.

Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.

Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION