Outlines

This application note is a reference material for developing products using I2C interface (I2C) functions of M3H Group (2).

This document helps the user check operation of the product and develop its program.

Target sample program: I2C_EEPROM
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 ......................................................................................................... 6
8. Outline of I²C Interface function ..................................................................................................... 7
  8.1. Clock Supply ................................................................................................................................. 7
9. Sample Program ............................................................................................................................... 8
  9.1. Initialization ................................................................................................................................ 8
  9.2. Sample program main operation ................................................................................................. 8
  9.3. Output Example of Sample Program ............................................................................................ 9
    9.3.1. Setting Example of Terminal Software .........................................................................................10
  9.4. Operating Flow of Sample Program ........................................................................................... 11
10. Precaution ..................................................................................................................................... 19
11. Revision History ............................................................................................................................ 19
RESTRICTIONS ON PRODUCT USE ................................................................................................ 20
1. Preface

This sample program is used to check the operation of I2C communication function. TMPM3HQ writes to I2C EEPROM and read from I2C EEPROM is executed by terminal software on the host PC via USB-UART.

Structure diagram of Sample program
2. Reference Document

- Datasheet
  TMPM3H group (2) datasheet Rev2.0 (Japanese edition)
- Reference manual
  I2C interface (I2C-B) Rev2.0 (Japanese edition)
- Other reference document
  TMPM3H(2) 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>ch1</td>
<td>PA4 (I2C1SCL) PA5 (I2C1SDA)</td>
<td>I2C mode</td>
</tr>
<tr>
<td>Asynchronous</td>
<td>ch0</td>
<td>PA1 (UT0TXDA) PA2 (UT0RXD)</td>
<td>UART mode</td>
</tr>
</tbody>
</table>

4. Target Device

The target devices of application note are as follows.

| TMPM3HQFDFG | TMPM3HQFZFG | TMPM3HQFYFG |
| TMPM3HPFDFG | TMPM3HPFZFG | TMPM3HPFYFG |
| TMPM3HNFDFG | TMPM3HNFZFG | TMPM3HNFYFG |
| TMPM3HNFDDFG | TMPM3HNFZDFG | TMPM3HNFYDFG |
| TMPM3HMFDFG | TMPM3HMFZFG | TMPM3HMFYFG |

* This sample program operates on the evaluation board of TMPM3HQFDFG. If other function than the TMPM3HQ one is checked, it is necessary that CMSIS Core related files (C startup file and I/O header file) should be changed properly. The BSP related file is dedicated to the evaluation board (TMPM3HQ). If other function than the TMPM3HQ one is checked, the BSP related file should be changed properly.
5. Conditions for Correct Operation

Used microcontroller: TMPM3HQFDFG
Used board: TMPM3HQFDFG 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

For purchasing the board, refer to the following homepage. (http://www.chip1stop.com/)
6. Evaluation Board Setting

The following pin connections should be done on the evaluation board.

<table>
<thead>
<tr>
<th>CN12</th>
<th>Board function</th>
<th>Through hole No.</th>
<th>Through hole No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>USB UART conversion</td>
<td>53 : PA1</td>
<td>54 : USB_TXD</td>
</tr>
<tr>
<td></td>
<td>USB UART conversion</td>
<td>55 : PA2</td>
<td>56 : USB_RXD</td>
</tr>
<tr>
<td></td>
<td>I2C (SCL)</td>
<td>57 : PA4</td>
<td>58 : I2C1_SCL</td>
</tr>
<tr>
<td></td>
<td>I2C (SDA)</td>
<td>59 : PA5</td>
<td>60 : I2C1_SDA</td>
</tr>
</tbody>
</table>

7. Operation of Evaluation Board

Connect the PC with the USB_UART terminal of the evaluation board with a USB cable. After start-up the terminal software (Tera Term), performs communication setting on the terminal software. Press the reset button on the evaluation board. Communication starts according to command input. For details of command input operation, refer to the 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

Enter the command (write or read) on the terminal software.
In the case of the write command, the input character is saved in the I2C EEPROM.
In the case of the read command, it reads the data saved in the I2C EEPROM and displays it in 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 operation, shift to the main function and perform the following initialization:
  1: Initialization of BSP (Board Support Package)
  2: Initialization of variables
  3: Initialize the application
  4: Initialization of I2C for EEPROM
  5: Main control of sample program

After the above processing, perform the following operations on PC terminal software (Tera Term).
"Command>" is displayed on Tera Term. Enter write or read command according to the following format.

"write" command: The input character should be stored to I2C EEPROM (0x0 address).
"read" command: I2C EEPROM (0x0 address) is read and displayed on the Tera Term.

Command format:
write command
write X                X: Any character
read command           
read

<table>
<thead>
<tr>
<th>Used memory</th>
<th>I2C EEPROM: 24FC256-I/SN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used port</td>
<td>SCL: PA4, SDA: PA5</td>
</tr>
</tbody>
</table>
9.3. Output Example of Sample Program

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

```
command > write a
write data > a
command > read
read data > a
command >
```
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()

Creation

bsp_i2c

I2C_reset()

Register allocation

I2C

CG_FSYSENA
FSYSENAB00 (PortA) Enable
FSYSENAB12 (I2C) Enable

Port
I2C1SCL, I2C1SDA setting

I2C_frequency(100KHz)

I2C_get_clock_setting(Instance address)

I2C Initial value setting

result = EEPROM_i2c_init()->success

result = I2C_init()->success

I2C frequency(100KHz)

I2C_get_clock_setting(Instance address)

frequency(I2C_get_clock_setting()->

I2C_init(Instance address)

i2c_frequency(100KHz)

I2C Initial value setting

I2C reset(Instance address)

I2C_reset()->

I2C reset(Instance address)

i2c_init()

Port:
I2C1SCL, I2C1SDA setting

CG: FSYSENA
FSYSENAB00 (PortA) Enable
FSYSENAB12 (I2C) Enable

Creation

EEPROM_i2c_init()
"write" command processing

result: strcmp( )
In the case of "write":

EEPROM_i2c_Write Page
(Transmission data and transmission Byte count)

Transmission data generation
Byte0 : Address High Byte
Byte1 : Address Low Byte
Byte2~n : Data

i2c_write
(Instance handle, Slave address, Transmission data,
Transmission Byte count, and stop = 1)

Transmission Byte count = i2c_write(-):
Transmission Byte count = EEPROM_i2c_WritePage(-):

"write" command procedure

Terminal: Display
Write data is displayed.

i2c_eeprom
bep_i2c
i2c

i2c_start
i2c_stop
i2c_byte_write(Slave address)
i2c_byte_write(Transmission data)

opt
[stop=1]

Transmission Byte count = i2c_write(-):
Transmission Byte count = EEPROM_i2c_WritePage(-):
"read" command processing

1. **Transmission data generation**
   - **Byte0**: Address High Byte
   - **Byte1**: Address Low Byte

2. **Ref**: Refer to "i2c_write" part in "write" command procedure.

3. **Write count**:
   - Write count = i2c_write(-)

4. **Reception Byte count**:
   - Reception Byte count = EEPROM_i2c_ReadData(-)

5. **Result**:
   - strncmp()

6. **In the case of "read"**:
   - **i2c_read** (Instance handle, Slave address, Reception data, Reception Byte count, and stop = 0)

   - **Loop**: loop[1, 受信バイト数-1]

   - **ref**: i2c_stop

   - **ref**: i2c_byte_read(last=0)

   - **ref**: i2c_stop

   - **ref**: i2c_byte_read(last=1)

   - **ref**: i2c_stop

7. **Terminal display**
   - Read data is displayed.

---

**Notes**:
- "read" command processing
- "read" command procedure
- **EEPROM_i2c_Read Data** (Reception data and Byte count)
- **i2c_eeprom**
- **bsp_i2c**
- **i2c**

**Revision**:
- Rev 1.0

**Date**:
- 2018-05-17
I2C start procedure

```
i2c_start
```

```
result = i2c_start(); Successful.
```

Start = 1*
(Only Internal information update)
I2C write procedure

```
void i2c_byte_write(Instance handle and Transmission data)
{
    if (start = 1)
    {
        I2C_start_condition(Instance handle and Slave address);
        I2C_write_data(Instance handle and Transmission data);
    }
    else
    {
        I2C_int_status(Instance handle);
        I2C_clear_int_status(Instance handle);
        I2C_get_ack(Instance handle);
        ACK_result = I2C_byte_write()->ACK;
    }
    I2C_int_status(Instance handle);
    I2C_clear_int_status(Instance handle);
    I2C_get_ack(Instance handle);
    ACK_result = I2C_byte_write()->ACK;
}
```
I²C read procedure

```c
i2c_byte_read(Instance handle)

bsp_i2c

I2C_read_data(Instance handle)

received data=i2c_byte_read():

Loop[with I2C interrupt request]

I2C_int_status(Instance handle)

I2C_int_status=I2C_int_status():

I2C_write_data(Instance handle, Dummy data 0)

I2C_write_data():

I2C_clear_int_status(Instance handle)

I2C_clear_int_status():

I2C_write_data():
```

```c
I2C_read_data(Instance handle)

received data=i2c_read_data():
```
I2C stop procedure

```c
i2c_stop(Instance handle)
```

```
I2C_stop_condition(Instance handle)
```

```
I2C_status_busy(Instance handle)
```

```
i2c_stop(-)
```
10. Precaution

When using the sample program with CPU other than TMPM3HQ, 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>2018-05-17</td>
<td>-</td>
<td>First release</td>
</tr>
</tbody>
</table>
REQUIREMENTS 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 combustions 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, stocking 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

2018-05-17