M4G Group (1)
Application Note
I²C Interface
(I2C-B)
MASTER/SLAVE

Outlines

This application note is a reference material for developing products using I²C interface (I2C) Master/Slave functions of M4G group (1).
This document helps the user check operation of the product and develop its program.

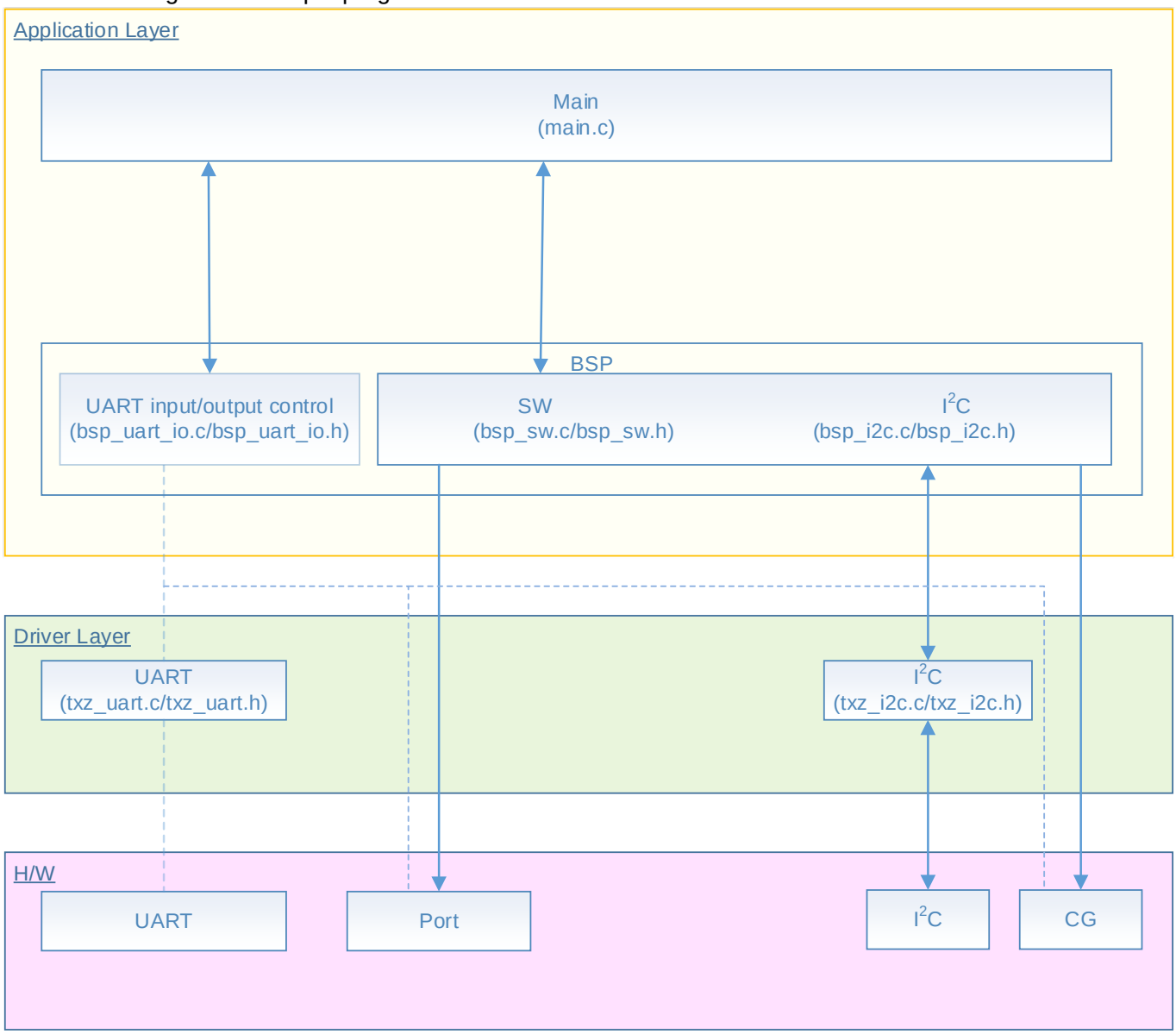
Target sample program: I2C_MASTER_SLAVE

Table of Contents

1. Preface

This sample program should be used to check the operation of the Master/Slave function in the I²C interface.
Two evaluation boards are used. The slave board is controlled by the command which is input to the master board via terminal software on PC.

Structure diagram of Sample program



2. Reference Document

- Datasheet
  TMPM4G group (1) datasheet Rev1.0 (Japanese edition)
- Reference manual
  I^2C Interface (I2C-B) Rev2.1 (Japanese edition)
  Asynchronous Serial Communication Circuit (UART-C) Rev3.0 (Japanese edition)
- Application note
  M4G group (1) Application Note Startup (CMSIS System & Clock Configuration) Rev1.0
- Other reference document
  TMPM4G (1) Group Peripheral Driver User Manual (Doxygen)

3. Function to Use

IP	Channel	Port	Function/Operation mode
Asynchronous Serial Communication Circuit	ch0	PE2 (UT0RXD) PE3 (UT0TXDA)	UART mode
I^2C Interface	ch3	PJ6 (I2C3SDA)	I^2C mode
		PJ7 (I2C3SCL)	

4. Target Device

The target devices of this application note are as follows;

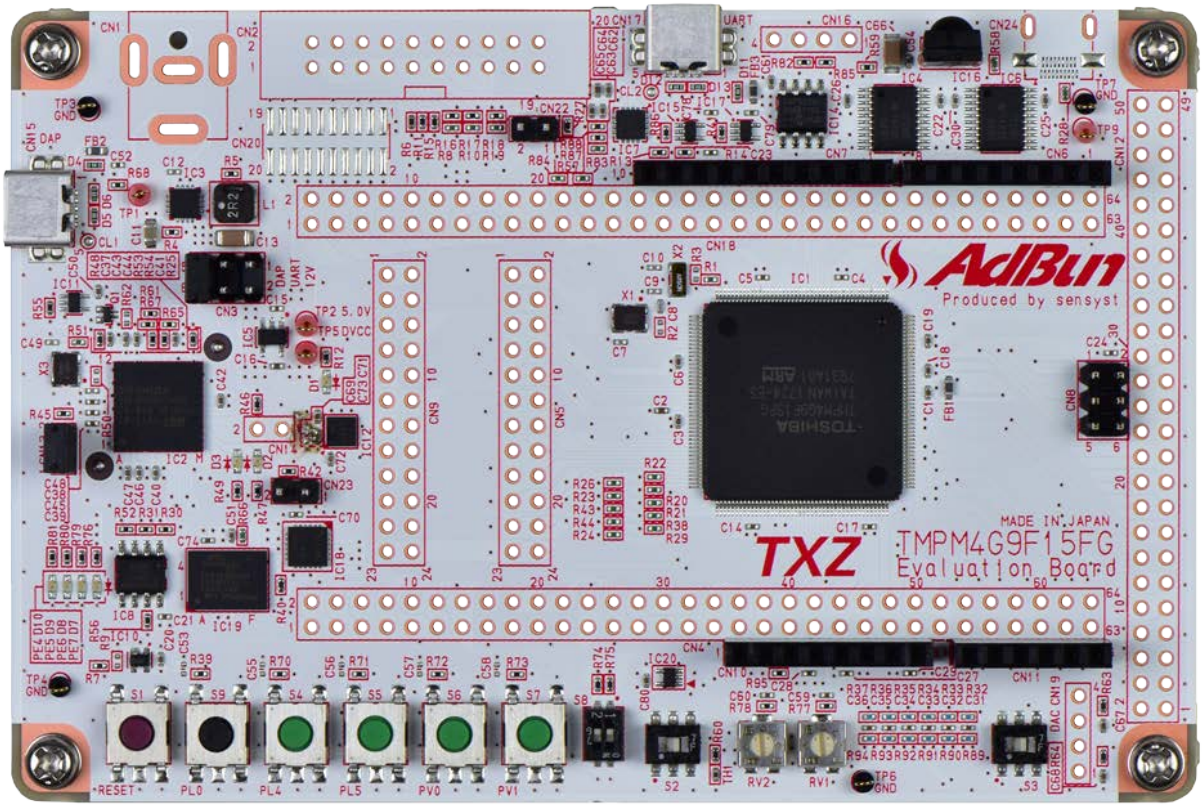
TMPM4G9F15FG	TMPM4G9F10FG	TMPM4G9FEFG	TMPM4G9FDFG
TMPM4G9F15XBG	TMPM4G9F10XBG	TMPM4G9FEXBG	TMPM4G9FDXBG
TMPM4G8F15FG	TMPM4G8F10FG	TMPM4G8FEFG	TMPM4G8FDFG
TMPM4G8F15XBG	TMPM4G8F10XBG	TMPM4G8FEXBG	TMPM4G8FDXBG
	TMPM4G7F10FG	TMPM4G7FEFG	TMPM4G7FDFG
	TMPM4G6F10FG	TMPM4G6FEFG	TMPM4G6FDFG

* This sample program operates on the evaluation board of TMPM4G9F15FG.
If other function than the TMPM4G9F15 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 (TMPM4G9F15). If other function than the TMPM4G9F15 one is checked, the BSP related file should be changed properly.

5. Operation Confirmation Condition

Used microcontroller	TMPM4G9F15FG
Used board	TMPM4G9F15FG Evaluation Board by 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	V1000

Evaluation board (TMPM4G9F15FG Evaluation Board) Top view

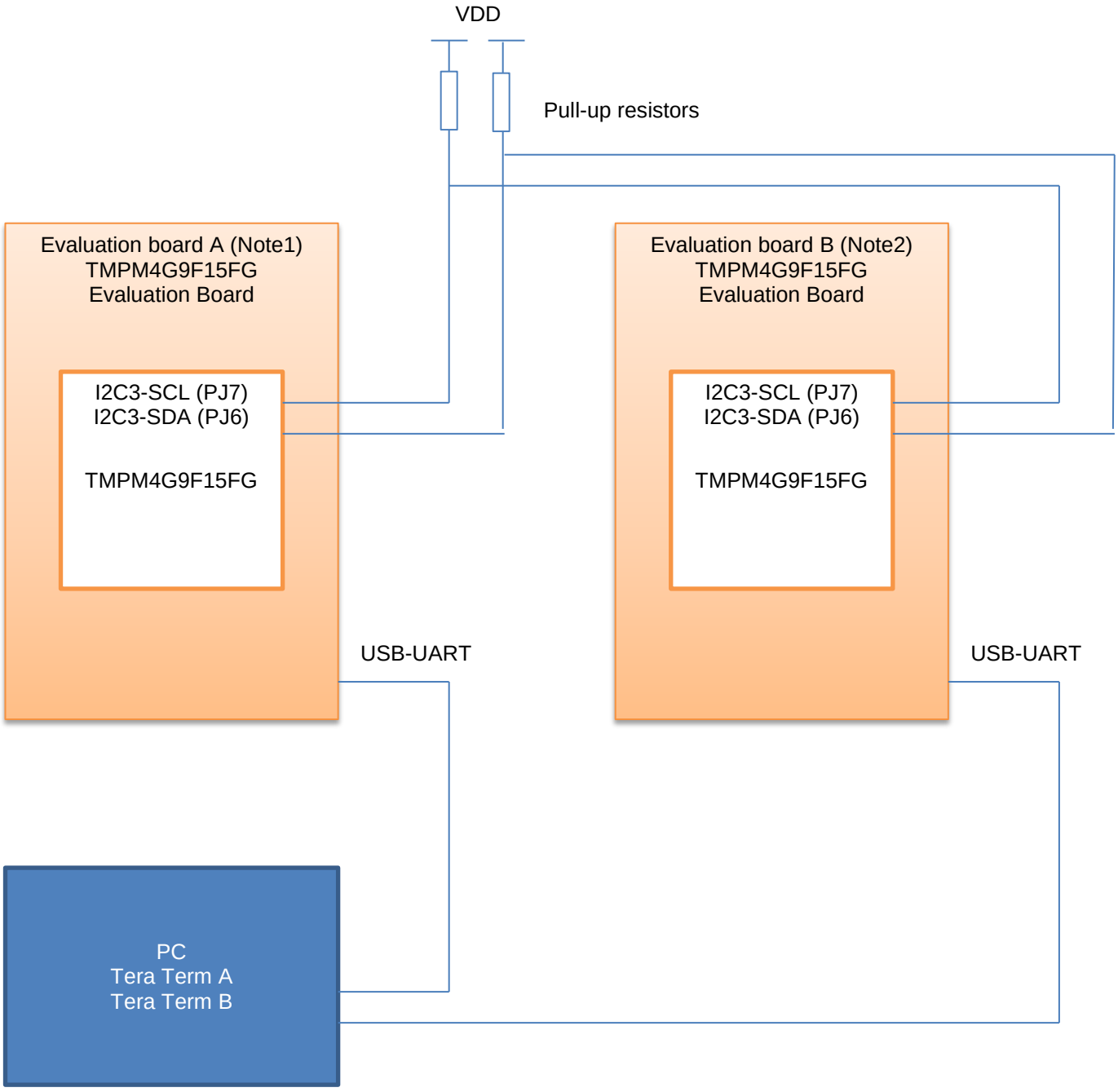


6. Evaluation Board Setting

Two evaluation boards are necessary for this project.

Evaluation boards:
Evaluation board A: TMPM4G9F15FG Evaluation Board
Evaluation board B: TMPM4G9F15FG Evaluation Board

These evaluation boards should be connected as shown in the following figure.



Note1: Evaluation board A

- The program which is generated in the project for TMPM4G9 should be written.
- Details of the connection:

CN5		
Board function	Through-hole No.	Through-hole No.
USB UART conversion	1: USB_UT_RX	2: PE2
USB UART conversion	3: USB_UT_TX	4: PE3

Through-hole connection on the board A:
[CN4] No. 39 on Evaluation board A and [CN5] No. 6 on Evaluation board A are connected.
[CN4] No. 41 on Evaluation board A and [CN5] No. 8 on Evaluation board A are connected.
Note) The connection to CN4 is done to use the pull-up resistor of the Port dip switch.
The Port dip switch should be set to OFF.

Note2: Evaluation board B

- The program which is generated in the project for TMPM4G9 should be written (the same program for Evaluation board A).
- Details of the connection:

CN5		
Board function	Through-hole No.	Through-hole No.
USB UART conversion	1: USB_UT_RX	2: PE2
USB UART conversion	3: USB_UT_TX	4: PE3

The interconnection between the boards:
[CN5] No. 6 on Evaluation board A and [CN5] No. 6 on Evaluation board B are connected.
[CN5] No. 8 on Evaluation board A and [CN5] No. 8 on Evaluation board B are connected.

7. Operation of Evaluation Board

Prepare two evaluation boards and connect each DAP connector and USB_UART connector to the PC.
When the sample program executes, both evaluation boards start up in the Master mode.
“slave” command should be input to one of the boards. Then, the board enters the Slave mode.
For the details of the operation of an input command, refer to the section “Sample Program Main Operation”.

8. Outline of I²C Function

The I²C can operate as a transceiver circuit of 1ch (SCL, SDA) in 1 unit circuit. The list of the functions is shown below.

Function classification (Note1)	Function	A Functional Description or the range
Transmission speed Control	prescaler dividing selection	It is dividing about a prescaler clock to 1/1, 1/2, 1/3 to 1/30, 1/31 and 1/32.
	Clock source	A selection setup of the HIGH/LOW time of SCL is possible in master mode.
	The maximum transfer rate	1Mbps (it corresponds to Fm+) (fsys = 8 to 200 MHz)
Communication Format	I²C bus format	Selection of Addressing/Data Free Format is possible. Selection of a master/slave is possible.
	Data length	1 to 8 bits
	acknowledge	The existence of acknowledging can be chosen.
	Start/stop condition	Generating of start/stop condition is possible.
	Slave address	Only a 7-bit addressing format. 2 sets of slave addresses can be set up. (1st/2nd Slave Address)
	General call	Detection of a general call is possible in slave mode.
Transmission and reception Control	Arbitration	Multi-master Clock synchronization Existence selection of Arbitration lost detection is possible.
	Repetitive start detection, generating	Detection of a repetitive start of a bus line (at the time of slave mode) and generating (at the time of master mode) are possible.
	Noise cancellation	Digital
Ganged control	Interruption	4 kinds (The completion interruption of transmission, Arbitration lost detection interruption, Bus free detection interruption, NACK reception detection interruption
	DMA request	A setup according to transmission and reception is possible.
	Software reset	Reset by the software of an I²C circuit is possible.
	Bus terminal state monitor function	The level monitor of SDA and a SCL pin
	Address match Wakeup function	Slave address match detection can use the release factor for the Low power consumption mode release.

Note1: It does not support HS (High Speed) mode, 10-bit addressing, and a START byte.
Note2: There is a function in which it cannot support depending on products, such as slope control, I/O correspondence at the time of the power supply OFF, an Input voltage (VIH/VIL), and an Output voltage (VOL=0.4V, VDD>2V).Please refer to the "Product Information" of the reference manual for details.

9. Sample Program

One of the evaluation boards can be set to the Slave mode by the command input on the terminal software. Then the Master board can control the Slave board.

9.1. Initialization

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

9.2. Sample Program Main Operation

The BSP (Board Support Package) is initialized.
As initialization of the application software, the UART initialization and the UART interrupt enable are done.
The configuration procedure of the I^2C interface is done.
The transmission and reception data of master and slave are initialized.
The setting values of the I^2C interface are initialized.
The sample program waits for the input of a command to the terminal software.
The I^2C master function or the I^2C slave function in the microcontroller can be executed by the input of a proper character according to the command format.
The master mode can be changed to the slave mode by a corresponding command. The command can be accepted only in the master mode.

Command format

"command[␣parameter]"

The parameter depends on a command.

Command	Parameter	Input example	note
write	non	write	The initial data is transmitted to the address of the initial value.
	address	write B0	The initial data is transmitted to the address B0.
	address+data	write B011223344	The setting data is transmitted to address B0.
read	non	read	Send the initial value from the setting address and receive the initial value.
	address	read B0	Send the initial value from the setting address and receive the initial value.
	address+data	read B05566	Send the setting value from the setting address and receive the initial value.
slave	non	slave	Set to slave mode.
	address	slave B0	Set slave mode and slave address.

Note: Input data is a hexadecimal number. “12” should be input for “0x12”, as an example.

Other command example

"w" command
"w" is the same as the command of "write" without any arguments.

"r" command
"r" is the same as the command of "read" without any arguments.

9.3. Output Example of Sample Program

When the sample program executes, the command input and the result are displayed as shown in the following figure.

[Basic log information]
"I2C3" shows the used channel in the I^2C interface.
"sa B0" shows the reception-wait Slave address.
"tx[ ]" and "rx[ ]" show a transmission data and a reception data, respectively.

Example of Master output log

```
I2C TEST - I2C3
 -----------------
| I2C master mode |
 -----------------
command >
```

```
command > write
master
sa      B0
tx[0] 00
tx[1] 01
tx[2] 02
tx[3] 03
command >
```

```
command > read
master
sa      B0
tx[0] 00
tx[1] 01
rx[0] 80
rx[1] 81
command >
```

Example of Slave output log

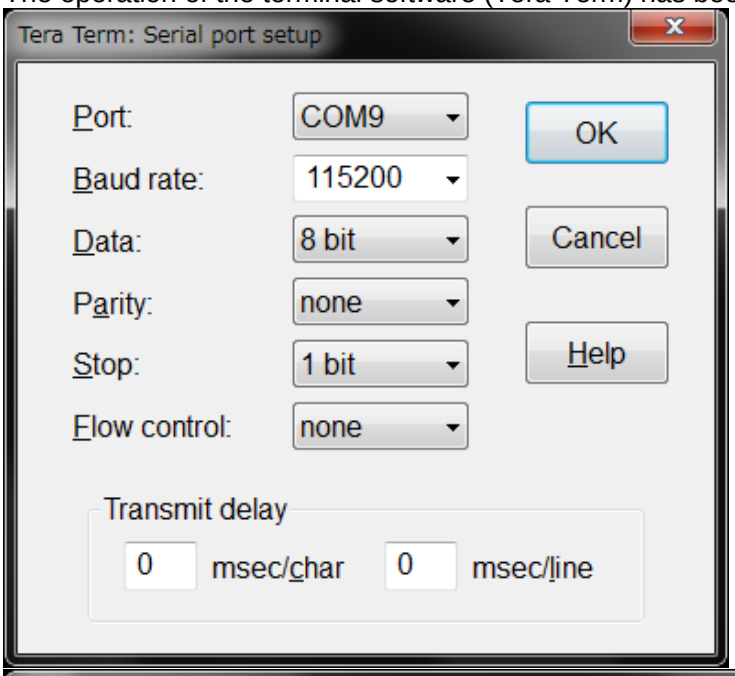
```
I2C TEST - I2C3
 -----------------
| I2C master mode |
 -----------------
command > slave
 ----------------
| I2C slave mode |
 ----------------
slave
sa      B0
```

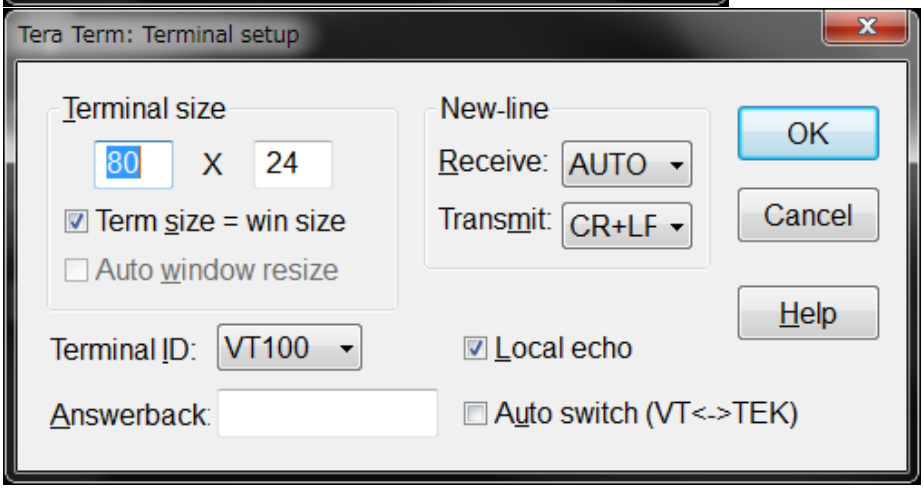
```
slave
sa      B0
rx[0] 00
rx[1] 01
rx[2] 02
rx[3] 03
slave
sa      B0
```

```
slave
sa      B0
rx[0] 00
rx[1] 01
tx[0] 80
tx[1] 81
slave
sa      B0
```

9.3.1. Setting Example of Terminal Software

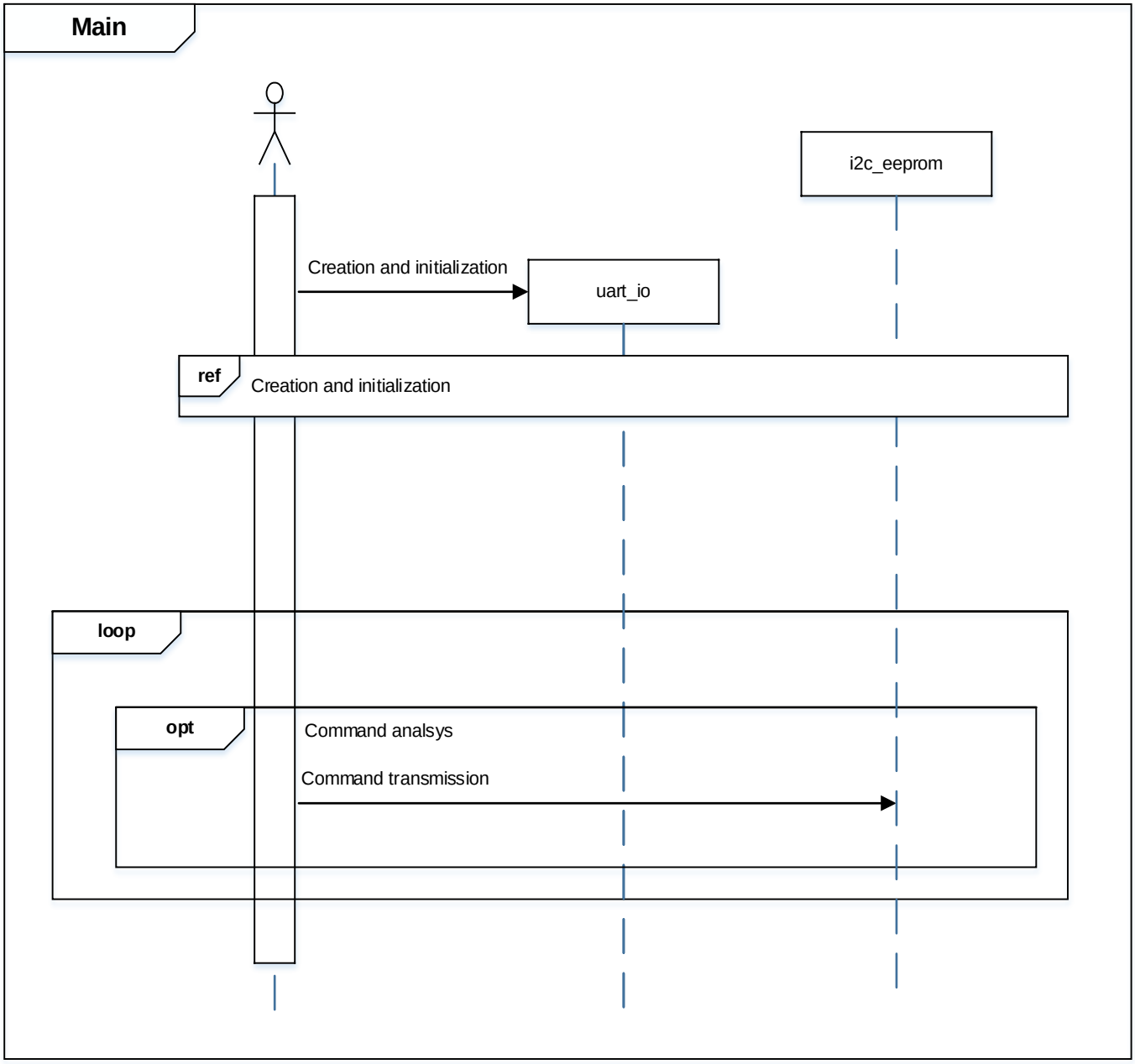
The operation of the terminal software (Tera Term) has been checked with the following settings.

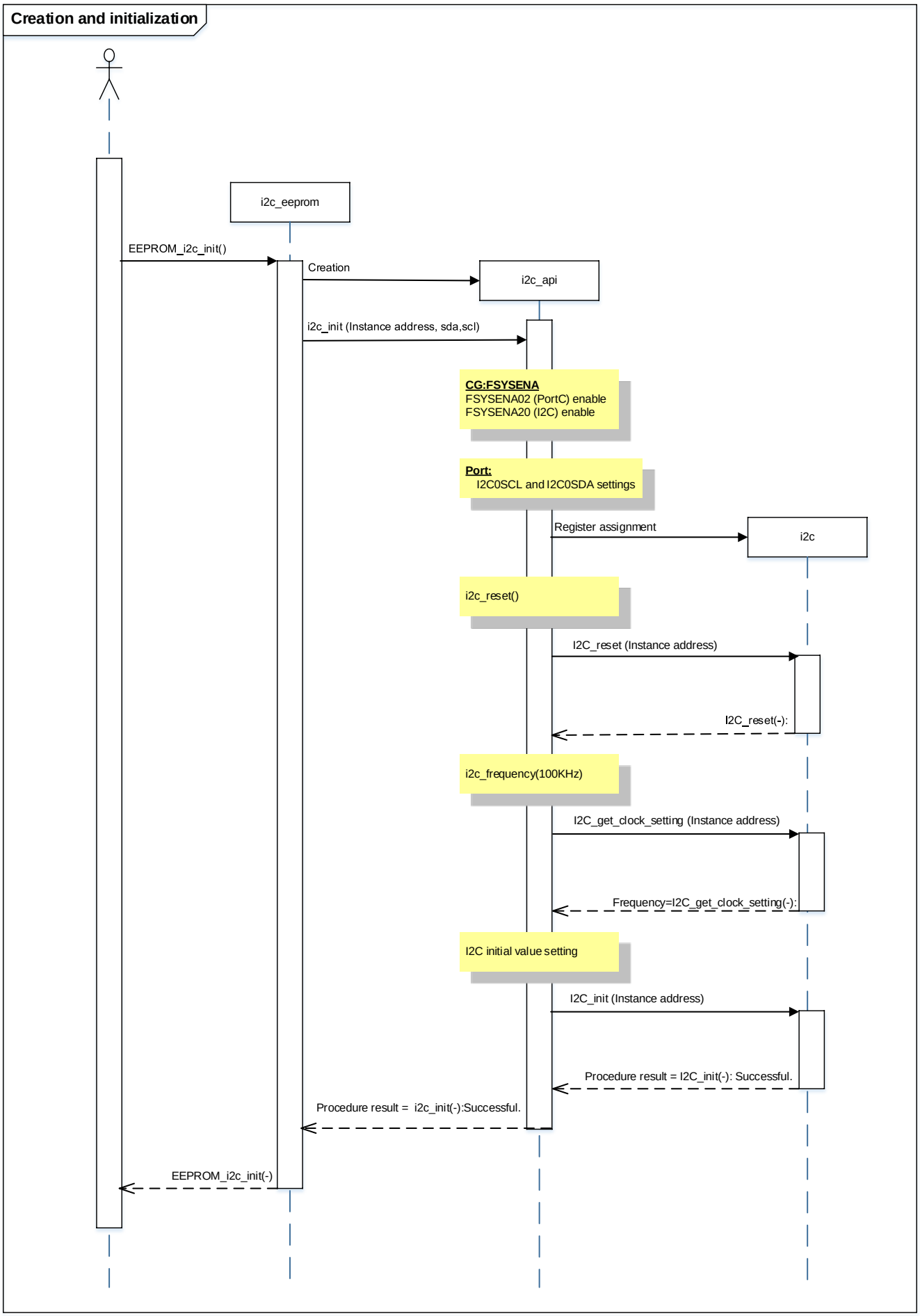


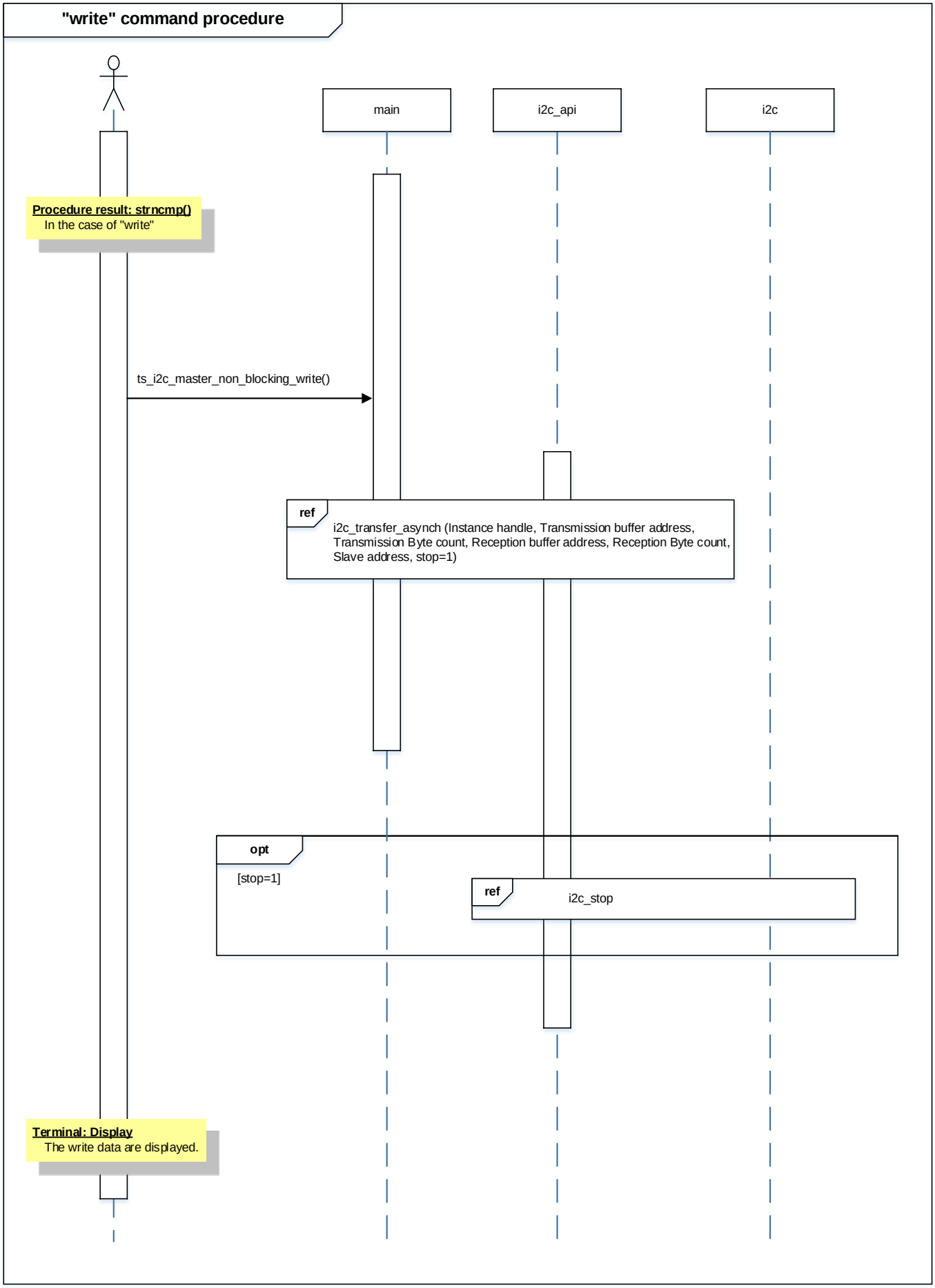


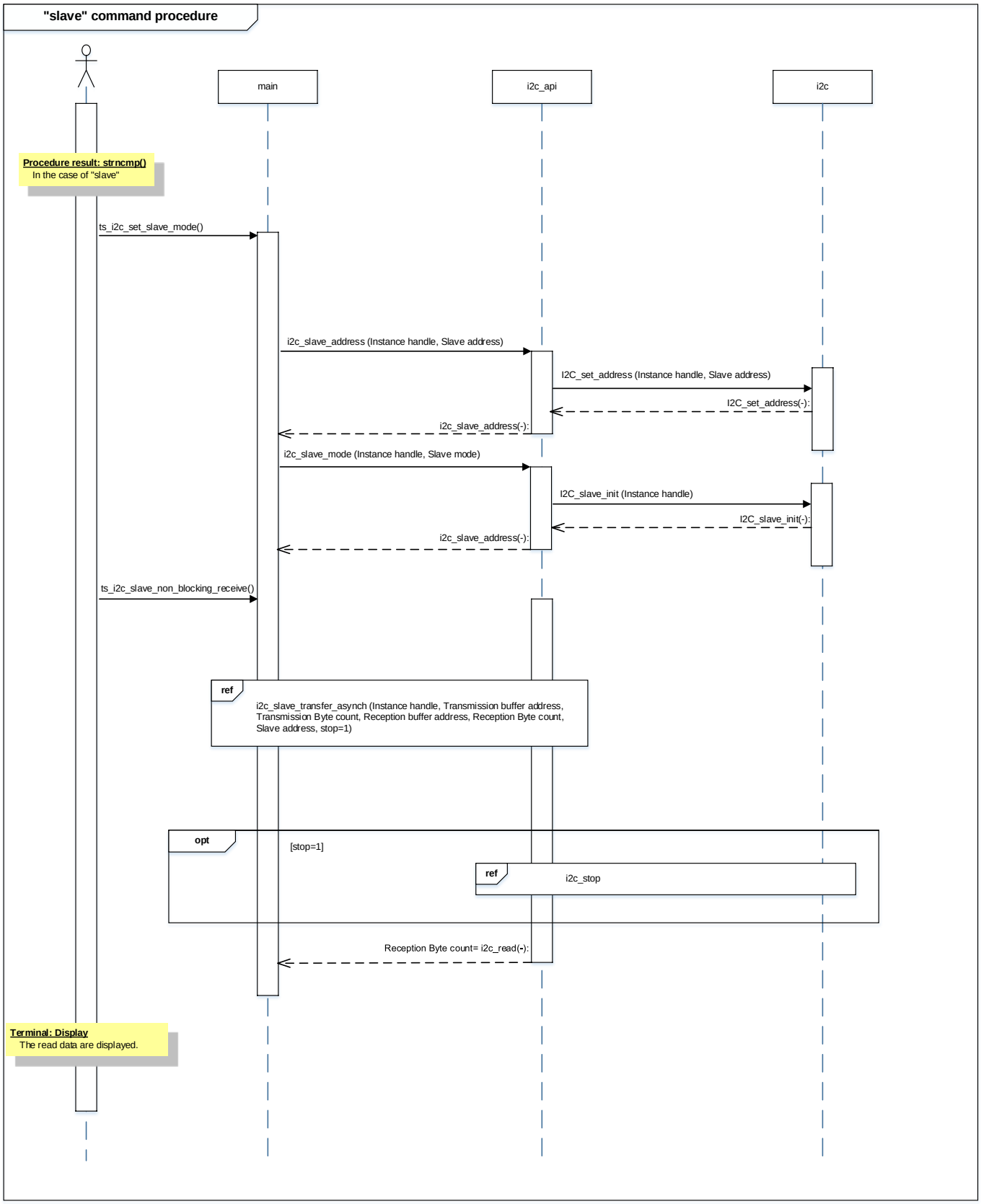
9.4. Operating Flow of Sample Program

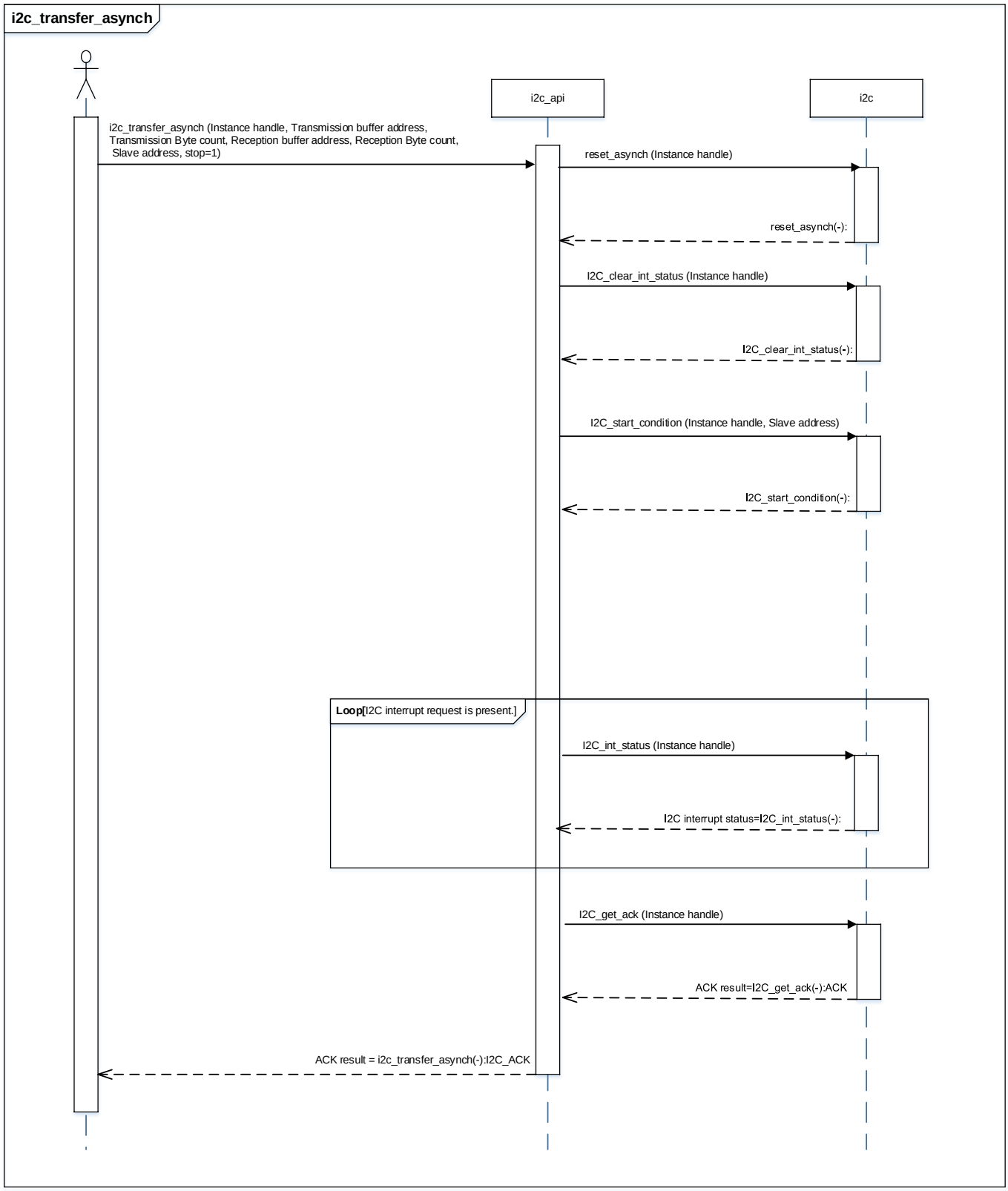
The basic operating flows of the sample program are shown in the following;

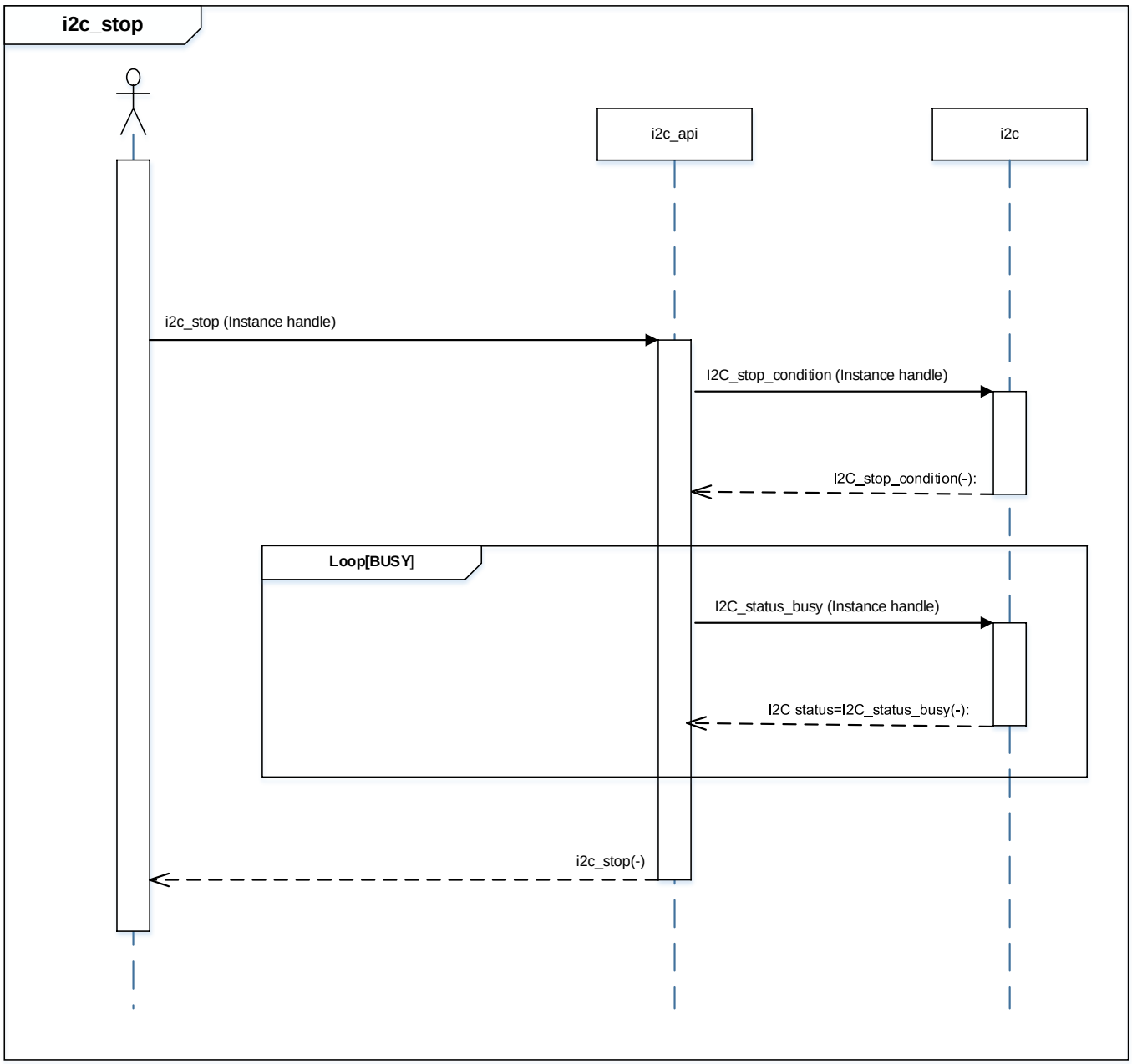



Creation and initialization
i2c_eeprom
EEPROM_i2c_init()
Creation
i2c_api
i2c_init (Instance address, sda,scl)
CG:FSYSENA
FSYSENA02 (PortC) enable
FSYSENA20 (I2C) enable
Port:
I2C0SCL and I2C0SDA settings
Register assignment
i2c
i2c_reset()
I2C_reset (Instance address)
I2C_reset(-):
i2c_frequency(100KHz)
I2C_get_clock_setting (Instance address)
Frequency=I2C_get_clock_setting(-):
I2C initial value setting
I2C_init (Instance address)
Procedure result = I2C_init(-): Successful.
Procedure result = i2c_init(-):Successful.
EEPROM_i2c_init(-)


"write" command procedure
main
i2c_api
i2c
Procedure result: strncmp()
In the case of "write"
ts_i2c_master_non_blocking_write()
ref
i2c_transfer_asynch (Instance handle, Transmission buffer address, Transmission Byte count, Reception buffer address, Reception Byte count, Slave address, stop=1)
opt
[stop=1]
ref
i2c_stop
Terminal: Display
The write data are displayed.


"slave" command procedure
main
i2c_api
i2c
Procedure result: strncmp()
In the case of "slave"
ts_i2c_set_slave_mode()
i2c_slave_address (Instance handle, Slave address)
I2C_set_address (Instance handle, Slave address)
I2C_set_address(-):
i2c_slave_address(-):
i2c_slave_mode (Instance handle, Slave mode)
I2C_slave_init (Instance handle)
I2C_slave_init(-):
i2c_slave_address(-):
ts_i2c_slave_non_blocking_receive()
ref
i2c_slave_transfer_asynch (Instance handle, Transmission buffer address, Transmission Byte count, Reception buffer address, Reception Byte count, Slave address, stop=1)
opt
[stop=1]
ref
i2c_stop
Reception Byte count= i2c_read(-):
Terminal: Display
The read data are displayed.


i2c_transfer_asynch
i2c_api
i2c
i2c_transfer_asynch (Instance handle, Transmission buffer address, Transmission Byte count, Reception buffer address, Reception Byte count, Slave address, stop=1)
reset_asynch (Instance handle)
reset_asynch(-):
I2C_clear_int_status (Instance handle)
I2C_clear_int_status(-):
I2C_start_condition (Instance handle, Slave address)
I2C_start_condition(-):
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
ACK result = i2c_transfer_asynch(-):I2C_ACK


i2c_stop
i2c_api
i2c
i2c_stop (Instance handle)
I2C_stop_condition (Instance handle)
I2C_stop_condition(-):
Loop[BUSY]
I2C_status_busy (Instance handle)
I2C status=I2C_status_busy(-):
i2c_stop(-)

10. Precaution

When using the sample program with CPU other than TMPM4G9F15, please check operation sufficiently.

11. Revision History

Rev	Date	Page	Description
1.0	2018-12-10	—	First release

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, lifesaving and/or life supporting 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, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- 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.**