Outlines

This application note is a reference material for developing products using ISP (In-System-Programming) of M3H group (2). This document helps the user check operation of the product and develop its program.

Target sample program: ISP_BOOT
Table of Contents

Outlines ............................................................................................................................................... 1
Table of Contents ................................................................................................................................ 2
1. Preface ........................................................................................................................................... 4
2. Outline of ISP .................................................................................................................................. 4
3. Reference Document ...................................................................................................................... 6
4. Function to Use ............................................................................................................................... 6
5. Target Device .................................................................................................................................. 6
6. Operation Confirmation Environment ............................................................................................. 7
7. Evaluation Board Setting .................................................................................................................. 8
8. Operation of Evaluation Board ....................................................................................................... 8
9. Sample Program ............................................................................................................................... 9
  9.1. Memory Map ...................................................................................................................................... 10
    9.1.1. Location Outline of ISP Firmware and Application Firmware .................................................. 11
    9.1.2. Location Outline of Flash Memory Operation Program at Execution of ISP Firmware ............... 12
  9.2. Build of ISP Firmware and Flash Memory Write ......................................................................... 12
    9.2.1. Build of ISP Firmware for Keil .................................................................................................. 12
    9.2.2. Build of ISP Firmware for IAR ................................................................................................. 12
  9.3. Build of Application Firmware .................................................................................................... 13
    9.3.1. Build of Application Firmware for Keil ..................................................................................... 13
    9.3.2. Build of Application Firmware for IAR ..................................................................................... 15
  9.4. Operation Flow ............................................................................................................................ 19
  9.5. Write and Execution of Application Firmware ............................................................................. 20
10. Building of GUI Tool (ISP_BOOT_GUI.exe) .................................................................................. 21
11. Points to Remember on Handling of Sample Programs ............................................................... 22
12. Revision History ............................................................................................................................. 22
REstrictions on product use .................................................................................................................. 23
Arm and Keil are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

All other company names, product, and service names mentioned herein may be trademarks or registered trademarks of respective companies.
1. Preface

This sample program is used to check the operation of the ISP (In-System-Programming) function. This ISP system is prepared as the example of programs which can re-write the built-in Flash memory in the microcontroller in the user boot mode at the application system, and which can operate in Windows at the host system.

2. Outline of ISP

The ISP is a method for updating the program region of the microcontroller in the system. The program to be updated is transferred to the application system from host system side.

Whole system outline of ISP

Application system

It means the final product which incorporates a microcontroller, and a board or module which implements a microcontroller.

When the application system is designed, a communication function, operating voltage, and the time to write should be concerned for writing with on-board to the Flash memory of the microcontroller.

The features required for the application program are as follows.

・Communication method with a host system
・Transfer program to the RAM
  (This program is required when the erase / write program is prepared beforehand in the Flash memory of the application system.)
・Erase / write program (when the program is prepared beforehand in the microcontroller of the application system)

The above programs except communication method should be planned that they are implemented beforehand to the microcontroller, or transferred from the host.

Host system

It means other microcontrollers, SoC, and PC or devices which is installed Windows, Linux, and others.

Programs and data to be updated are transferred from the host system to the application system.

The communication method with the application system is required for the host system.
The features required for the host system are as follows.

・Communication method with the application system
・Erase / write operation of the Flash memory (when the erase / write operation is transferred from the host system without preparation of the application system)

Additionally, the application system and the host system are required to ensure the security, respectively.
3. Reference Document

- Datasheet
  TMPM3H group (2) datasheet Rev4.1
- Reference Manual
  Flash Memory (Flash512_32-A ) Rev2.0
  Asynchronous Serial Communication Circuit ( UART-C ) Rev3.1
  Input and Output Ports (PORT-M3H(2)) Rev4.0
- Other reference document
  TMPM3H(2) Group Peripheral Driver User Manual (Doxygen) v1.0.10.0

4. 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>Flash memory</td>
<td>—</td>
<td>—</td>
<td>Code flash</td>
</tr>
<tr>
<td>Asynchronous serial</td>
<td>ch0</td>
<td>PA1 (UT0TXDA)</td>
<td>UART mode</td>
</tr>
<tr>
<td>communication circuit</td>
<td></td>
<td>PA2 (UT0RXD)</td>
<td></td>
</tr>
<tr>
<td>Input and output ports</td>
<td>—</td>
<td>PK4 (Output Port)</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PK5 (Output Port)</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PK6 (Output Port)</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PK7 (Output Port)</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PV0 (Input Port)</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PV1 (Input Port)</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PV2 (Input Port)</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>PV3 (Input Port)</td>
<td>Input</td>
</tr>
</tbody>
</table>

The PK6, PV0, PV1, and PV2 are used in the sample program (GPIO_LED) of the application system.

5. Target Device

The target devices of this application note are as follows;

<table>
<thead>
<tr>
<th>TMPM3HQFDFG</th>
<th>TMPM3HQFZFG</th>
<th>TMPM3HQFYFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMPM3HPFDFG</td>
<td>TMPM3HPFZFG</td>
<td>TMPM3HPFYFG</td>
</tr>
<tr>
<td>TMPM3HNFDFG</td>
<td>TMPM3HNFZFG</td>
<td>TMPM3HNFYFG</td>
</tr>
<tr>
<td>TMPM3HNFDG</td>
<td>TMPM3HNFDG</td>
<td>TMPM3HNFDG</td>
</tr>
<tr>
<td>TMPM3HMFDUG</td>
<td>TMPM3HMFDUG</td>
<td>TMPM3HMFDUG</td>
</tr>
<tr>
<td>TMPM3HLFDUG</td>
<td>TMPM3HLFDUG</td>
<td>TMPM3HLFDUG</td>
</tr>
</tbody>
</table>

* 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 (the startup file and I/O header file) should be changed properly.
  Additionally, the name of microcontroller which is set to the project should be changed.
  The BSP related file is dedicated to the evaluation board (TMPM3HQFDFG). If other function than the TMPM3HQFDFG one is checked, the BSP related file should be changed properly.
6. Operation Confirmation Environment

<table>
<thead>
<tr>
<th>Used microcontroller</th>
<th>TMPM3HQFDHG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used board</td>
<td>TMPM3HQFDHG Evaluation Board by Sensyst</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>IAR Embedded Workbench for ARM 8.22.2</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Arm® Keil® MDK Version 5.24.2.0</td>
</tr>
<tr>
<td>Terminal software</td>
<td>Tera Term V4.96</td>
</tr>
<tr>
<td>Sample program</td>
<td>v1.2.0</td>
</tr>
<tr>
<td>ISP_BOOT Sample program</td>
<td>v1.0.0</td>
</tr>
</tbody>
</table>

For purchasing the board, refer to the following homepage. (http://www.chip1stop.com/)
7. 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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CN5</th>
<th>Board function</th>
<th>Through-hole No.</th>
<th>Through-hole No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED0 (D10)</td>
<td></td>
<td>33: PORT_LED0</td>
<td>34: PK4</td>
</tr>
<tr>
<td>LED1 (D9)</td>
<td></td>
<td>35: PORT_LED1</td>
<td>36: PK5</td>
</tr>
<tr>
<td>LED2 (D8)</td>
<td></td>
<td>37: PORT_LED2</td>
<td>38: PK6</td>
</tr>
<tr>
<td>LED3 (D7)</td>
<td></td>
<td>39: PORT_LED3</td>
<td>40: PK7</td>
</tr>
<tr>
<td>Push switch0 (S4)</td>
<td></td>
<td>49: PORT_PSW0</td>
<td>50: PV0</td>
</tr>
<tr>
<td>Push switch1 (S5)</td>
<td></td>
<td>51: PORT_PSW1</td>
<td>52: PV1</td>
</tr>
<tr>
<td>Push switch2 (S6)</td>
<td></td>
<td>53: PORT_PSW2</td>
<td>54: PV2</td>
</tr>
<tr>
<td>Push switch3 (S7)</td>
<td></td>
<td>55: PORT_PSW3</td>
<td>56: PV3</td>
</tr>
</tbody>
</table>

8. Operation of Evaluation Board

The PC and the USB_UART connector in the evaluation board should be connected with a USB cable. The reset is asserted while the Flash memory rewrite button is pushed down. After the LED1 lights, the GUI tool (ISP_BOOT.exe) should be started up and the communication setting is done.
9. Sample Program

This ISP sample program is supposed to be used as a part of the TMPM3HQ sample program. The ISP sample program should be placed in a sub-folder under the folder of the TMPM3HQ sample program, as shown in the following.

Folder name of ISP sample program: ISP_BOOT

Brief of the folder configuration of the TMPM3HQ sample program:

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMPM3Hy_v1.2.0</td>
<td>Sample Program Root</td>
</tr>
<tr>
<td>TMPM3Hy_v1.2.0\Libraries</td>
<td>CMSIS-CORE, Hardware Abstraction Layer(HAL)</td>
</tr>
<tr>
<td>TMPM3Hy_v1.2.0\Project</td>
<td>Sample Program</td>
</tr>
</tbody>
</table>

Brief of the folder configuration of the ISP sample program (ISP_BOOT is in Project\Examples.)

<table>
<thead>
<tr>
<th>Folder name</th>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project\Examples\ISP_BOOT</td>
<td>ISP_BOOT.ewp</td>
<td>ISP firmware production project file (for IAR)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT</td>
<td>ISP_BOOT.ewt</td>
<td>ISP firmware production project file (for Keil)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT</td>
<td>ISP_BOOT.uvoptx</td>
<td>ISP firmware production project file (for Keil)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT</td>
<td>ISP_BOOT.uvprojx</td>
<td>ISP firmware production link file (for IAR)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT</td>
<td>ISP_BOOT.icf</td>
<td>ISP firmware production link file (for Keil)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOTbin</td>
<td>LED_GPIO_IAR.bin</td>
<td>Reference GPIO_LED firmware (IAR building version)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOTbin</td>
<td>LED_GPIO_KEIL.bin</td>
<td>Reference GPIO_LED firmware (Keil building version)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\ISP_BOOT_GUI\bin\Debug</td>
<td>ISP_BOOT_GUI.exe (This file doesn't include this sample PJ.)</td>
<td>ISP-dedicated GUI tool (for application firmware write)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\ISP_BOOT_GUI</td>
<td>Other than ISP_BOOT_GUI.exe</td>
<td>ISP_BOOT_GUI.exe production (C# code group: the individual description is omitted.)</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>main.c</td>
<td>ISP main process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>main.h</td>
<td>ISP main process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>Flash_Erase.c</td>
<td>Flash write process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>Flash_Erase.h</td>
<td>Flash write process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>Flash_Write.c</td>
<td>Flash erase process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>Flash_Write.h</td>
<td>Flash erase process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>com_control.c</td>
<td>UART communication process code</td>
</tr>
<tr>
<td>Project\Examples\ISP_BOOT\src</td>
<td>com_control.h</td>
<td>UART communication process code</td>
</tr>
</tbody>
</table>
9.1. Memory Map

The sub-section describes the location outline of the ISP firmware and the application firmware, and the Flash memory process at the execution of the ISP firmware.

The ISP is the program which can rewrite the built-in Flash memory in the microcontroller in the user boot mode. The location of the ISP programmed in the Flash memory starts with the first address in the memory. The region should not be erased nor be rewritten when the ISP is used. The application firmware which is rewritten should be located after the address region of the ISP. The addresses of the firmware should be set correctly at build.
9.1.1. Location Outline of ISP Firmware and Application Firmware

After the ISP firmware is written, the application firmware should be located in the address of 0x00002000 and after.
9.1.2. Location Outline of Flash Memory Operation Program at Execution of ISP Firmware

The program of the process of the Flash memory operation (erase/write) is copied to the RAM from the Flash memory when the ISP firmware is executed. The Flash memory operation is executed by the copied program on the RAM.

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>ISP main (Including UART communication)</td>
</tr>
<tr>
<td>0x00001000</td>
<td>Flash write program (Not executed.)</td>
</tr>
<tr>
<td>0x00002000</td>
<td>Flash erase program (Not executed.)</td>
</tr>
<tr>
<td>0x20000000</td>
<td>RAM region</td>
</tr>
<tr>
<td>0x20001000</td>
<td>Flash write program (Executed.)</td>
</tr>
<tr>
<td>0x20002000</td>
<td>Flash erase program (Executed.)</td>
</tr>
<tr>
<td>0x00010000</td>
<td>RAM (64 KB)</td>
</tr>
<tr>
<td>0x00018000</td>
<td>Flash memory (512 KB)</td>
</tr>
<tr>
<td>0x0007FFFF</td>
<td>Application</td>
</tr>
</tbody>
</table>

The programs are copied to the RAM from the Flash memory when the ISP is used.

9.2. Build of ISP Firmware and Flash Memory Write

The projects for Keil and IAR are prepared in the ISP firmware. The build of the project and the Flash memory write should be done according to the integrated development environment which is used for the application development. The ISP firmware is supposed to use an 80-MHz clock for operation.

9.2.1. Build of ISP Firmware for Keil

ISP_BOOT.uvprojx in the ISP_BOOT folder should be started up. The project should be built and the firmware is generated. The generated firmware should be programmed to the Flash memory. When the programming to the Flash memory succeeds, the build of the ISP firmware and the programming operation complete.

9.2.2. Build of ISP Firmware for IAR

ISP_BOOT.eww in the ISP_BOOT folder should be started up. The project should be built and the firmware is generated. The generated firmware should be programmed to the Flash memory. When the programming to the Flash memory succeeds, the build of the ISP firmware and the programming operation complete.
9.3. Build of Application Firmware

The ISP firmware is written to the Flash memory addresses 0x00000000 to 0x00001FFF. So the application firmware should be located in the address region which starts with 0x00002000. And the vector offset register should be modified to change the start address of the code. As the modification of the vector offset register, the execution code should be added to the application firmware. The addition code should be located before the use of an interrupt, for example, inside the “SystemInit” function for the start process, at the head of the “main” function, and others.

Modification example of the vector offset register (in the case of CMSIS description): SCB->VTOR = 0x000002000;

After the reset, the application firmware is executed via the un-rewrite process of the ISP firmware. So the clock setting and others are kept as the settings of the ISP firmware (the 80-MHz clock and others). If the execution with the 10-MHz clock or others generated by the internal oscillator is required for the application firmware, the appropriate code modification should be done explicitly.

The build of an application firmware for Keil and IAR are shown below. This example is the GPIO_LED operation.

9.3.1. Build of Application Firmware for Keil

LED_GPIO.uvprojx in the GPIO_LED folder should be started up.
The start address of the location of the application firmware and its size should be modified properly.
The setting should be done on the following window after selecting [Flash]->[Configure Flash Tools…]->[Target].

![Options for Target 'LED_GPIO']

IROM1:
Start; Change from 0x0 to 0x2000
Size: Change from 0x80000 to 0x7C000
It should be set to enable to generate the binary file (.bin) of the application firmware.
“fromelf.exe” is used to generate the binary file (.bin). “fromelf.exe” is an image conversion utility of the
ARM compiler tool chain.
The conversion from the object file (.axf) to the binary file (.bin) can be done by a command line using
“fromelf.exe”. There is another way. The binary file can be generated together with the build using the
following settings on the integrated development environment.

The setting should be done on the following window after selecting [Flash]->[Configure Flash
Tools…]->[User].

After the settings above are done, the build is executed. The generated binary file (.bin) is the binary file of
the target application firmware to which the ISP writes.
9.3.2. Build of Application Firmware for IAR

ISP_BOOT.eww in the GPIO_LED folder should be started up. The start address of the location of the application firmware and its size should be modified properly. The setting should be done on the following window after selecting [Project]->[Option]->[Linker]->[Config].

- Linker configuration file
  Check “Override default”.
  After that, click “Edit” button.
When [Edit] is clicked, [Linker configuration file editor] window is displayed. The values of [Vector Table] and [Memory regions] should be set for the application firmware.

After the values are set, [Save] should be clicked.

The following error message is displayed because the overwrite is not enabled on the default linker file ($TOOLKIT_DIR$\config\linker\Toshiba\TMPM3HQFDFG.icf). [OK] should be clicked.

The file name (LED_GPIO.icf) and the folder name for the stored data are displayed. [Save] should be clicked.
The linker setting file is modified as follows. Then [OK] should be clicked.
It should be set to enable to generate the binary file (.bin) of the application firmware.
The setting should be done on the following window after selecting [Project]->[Option]->[Output converter]->[Output].

- Linker configuration file
  Set linker configuration file that uses at application firmware.
- Output
  “Generate additional output” should be checked
  “Raw binary” should be set to “Output format”.
  “Override default” should be checked.
  The name of the binary file of the application firmware should be input (“LED_GPIO.bin” as an example).
9.4. Operation Flow

The basic operating flow of the ISP firmware is shown in the following as the sample program;

1. GUI tool (ISP_BOOT_GUI.exe)
   Click the Start communication button.
   - Error output?
     - Yes (LED0 lighting)
       - Return to Reset
     - No (LED0 lighting)
       - GUI tool (ISP_BOOT_GUI.exe)
         Click the Application transfer button.
         Write binary file (.bin) is selected.
   - Correct write completion?
     - Yes (LED3 lighting)
       - Application start-up
     - No
       - Return to Reset

The application program can be started up in the following three cases.
1. GUI tool (ISP_BOOT_GUI.exe)
   Click the Application activation button.
2. GUI tool (ISP_BOOT_GUI.exe)
   Click the System reset button.
3. Reset switch on the board is ON, or the power supply is ON after the power supply is OFF.
9.5. Write and Execution of Application Firmware

The GUI tool function which is used for the operating flow in the previous page is shown as follows. The operations should be done according to the operating flow of the sample program.

- Status at GUI tool start-up
  - COM port selection: After Communication baud rate (bps) is set, "Start Communication" should be clicked.
  - Setting parameters for the serial port are fixed to the following setting values.
    - Data: 8 bits
    - Parity: No parity
    - Stop bit: 1 bit
    - Flow control: No flow control

- Status after serial communication establishment
  - Click the Application Transfer button, and select the target binary file (.bin).
  - When the write is successful, the application can be started up directly with the following two cases.
    1. Direct startup without reset
       - Click the Application Activation button.
       - (In the case of the application firmware with using the interrupt, it has a restriction not to operate.)
    2. Reset at once and start up
       - System Reset

The start-up of the application firmware can be done by the reset (the start of the power supply or the reset assertion) on the evaluation board.
10. Building of GUI Tool (ISP_BOOT_GUI.exe)

ISP_BOOT_GUI.exe is configured with the C# language. The build environment is as follows:

Microsoft Visual Studio Professional 2017
Version 15.8.8
VisualStudio.15.Release/15.8.8+28010.2048
Microsoft .NET Framework
Version 4.7.03190

Visual C++ 2017 00370-20001-02825-AA159
Microsoft Visual C++ 2017

C# tool 2.9.0-beta8-63208-01

The source code which is used to build ISP_BOOT_GUI.exe is stored in the ISP_BOOT_GUI folder. It is possible to build an original GUI tool by modifying the source code.

It should be notified that Toshiba does not support the build of an original GUI tool.

The solution file (.sln) which is necessary at build is stored in the following folder.

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project\Examples\ISP_BOOT\ISP_BOOT_GUI</td>
<td>ISP_BOOT_GUI.sln</td>
<td>Microsoft Visual Studio Solution File</td>
</tr>
</tbody>
</table>
11. Points to Remember on Handling of Sample Programs

When using the sample program with other than Operation Confirmation Environment, please check operation sufficiently.

12. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2019-07-26</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, 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.