Self-diagnosis Program
Application Note
Basic Setting

Outlines

This application note describes the basic setting to use the self-diagnosis program.
This document should be referred when the self-diagnosis program is used by a TXZ series microcontroller.
This library supports the sample program with the peripheral driver of TXZ series enclosed.

Table of Contents

1. Preface

This application note is a document for the basic setting at the use of the self-diagnosis program.
This note describes the information about the settings, the operation environment, and others when the test is executed.
This library code should be used after it is added (overwritten) to the published sample program.

The explanation of this document uses the TMPM4K Group (1).
When the program is applied to a product, some appropriate modification may be necessary depending on the specifications of the product.

2. Operation Confirmation Environment

Used microcontroller	TMPM4K4AFYAUG
Integrated development environment	IAR Embedded Workbench for ARM 8.22.2
Integrated development environment	Arm® Keil® MDK Version 5.24.2.0

3. Setting

This explanation is an example of using TMPM4KxA sample program.
This library code should be used after it is added (overwritten) to the published TMPM4KxA sample program.

3.1. Tree Structure

The supplied tree is overwritten on the sample program v1.0.0, and it is expanded. The following tree format is given.

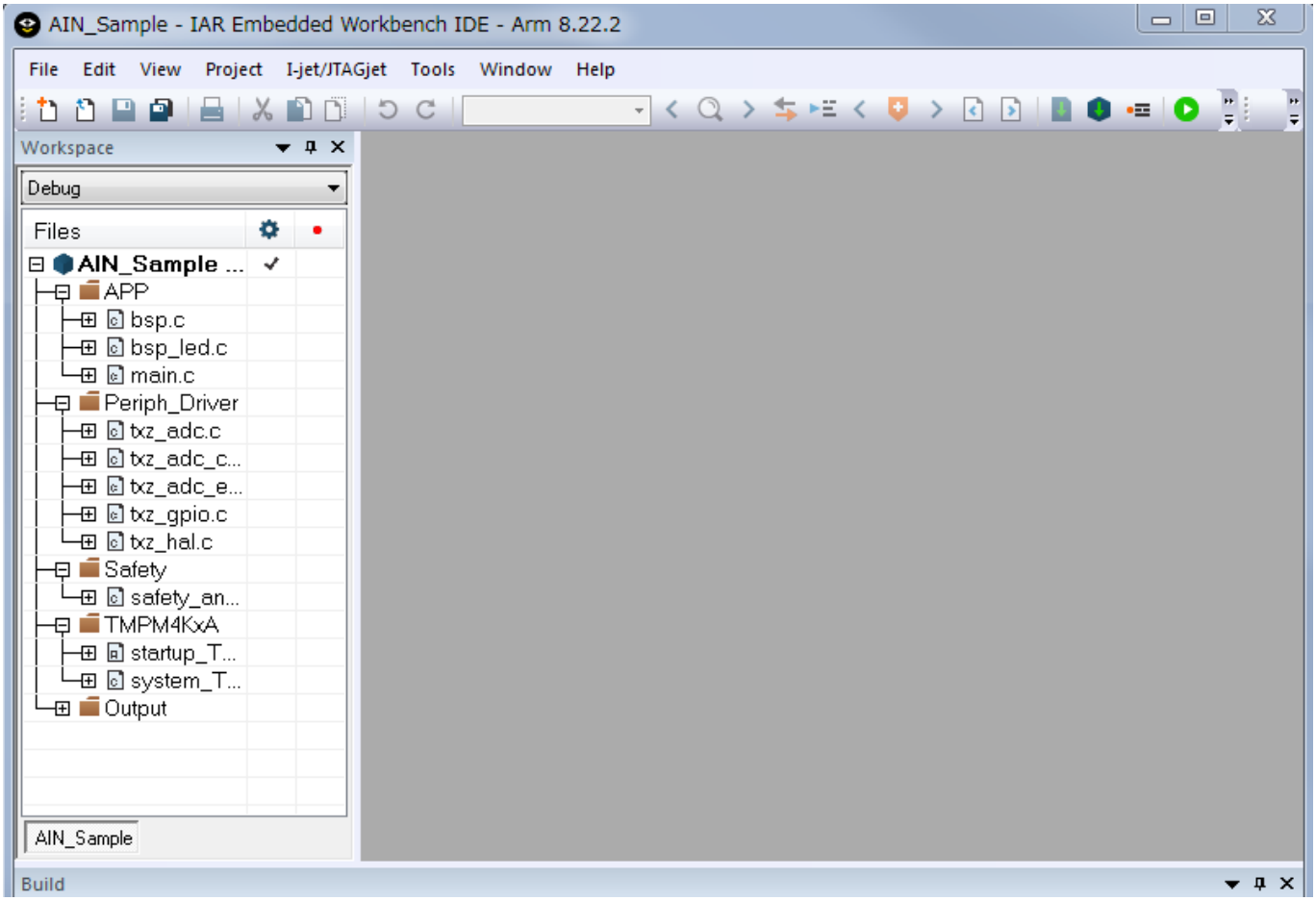
```
M4K4A_V100
├──Libraries
│   ├──CMSIS
│   │   └──startup : Including the start-up for CMSIS correspondence.
│   │       ├──arm
│   │       └──iar
│   ├──Doc
│   ├──Periph_driver : Including each driver code.
│   │   ├──inc
│   │   └──src
│   └──Safety : Including supplied self-diagnosis library code.
│       ├──inc
│       └──src
└──Project
    └──Examples
        ├──ADC
        ├──BSP : Including the BSP code.
        │   └──TMPM4K4A_eval
        │       └──src
        ├──CFC
        ├──CGRST
(snip)
        ├──Safety : Including the project which uses the self-diagnosis library.
        │   ├──AIN_Sample : Analog input sample project
        │   ├──CLOCKIRQ_Sample : Sample project for the test of the clock frequency and interrupt.
        │   ├──CPU_Sample : Sample project for the CPU registers and the PC
        │   ├──DIO_Sample : Digital IO sample project
        │   ├──MEMORY_Sample : Sample project for the RAM and the Flash memory.
        │   └──UART_Sample : Sample project for the UART communication.
        ├──T32A
        ├──TRM
        ├──TSPI
        ├──UART
```

Each function of the self-diagnosis library has been included in one of the sample projects. The operation of the function can be checked when the sample project should be built, be downloaded to the TMPM4K4 evaluation board, and be executed.
The colored parts are additional files for self-diagnosis.

3.2. Build of Project

<In the case of IAR Embedded Workbench>
• A project is built in the IAR Embedded Workbench as follows;

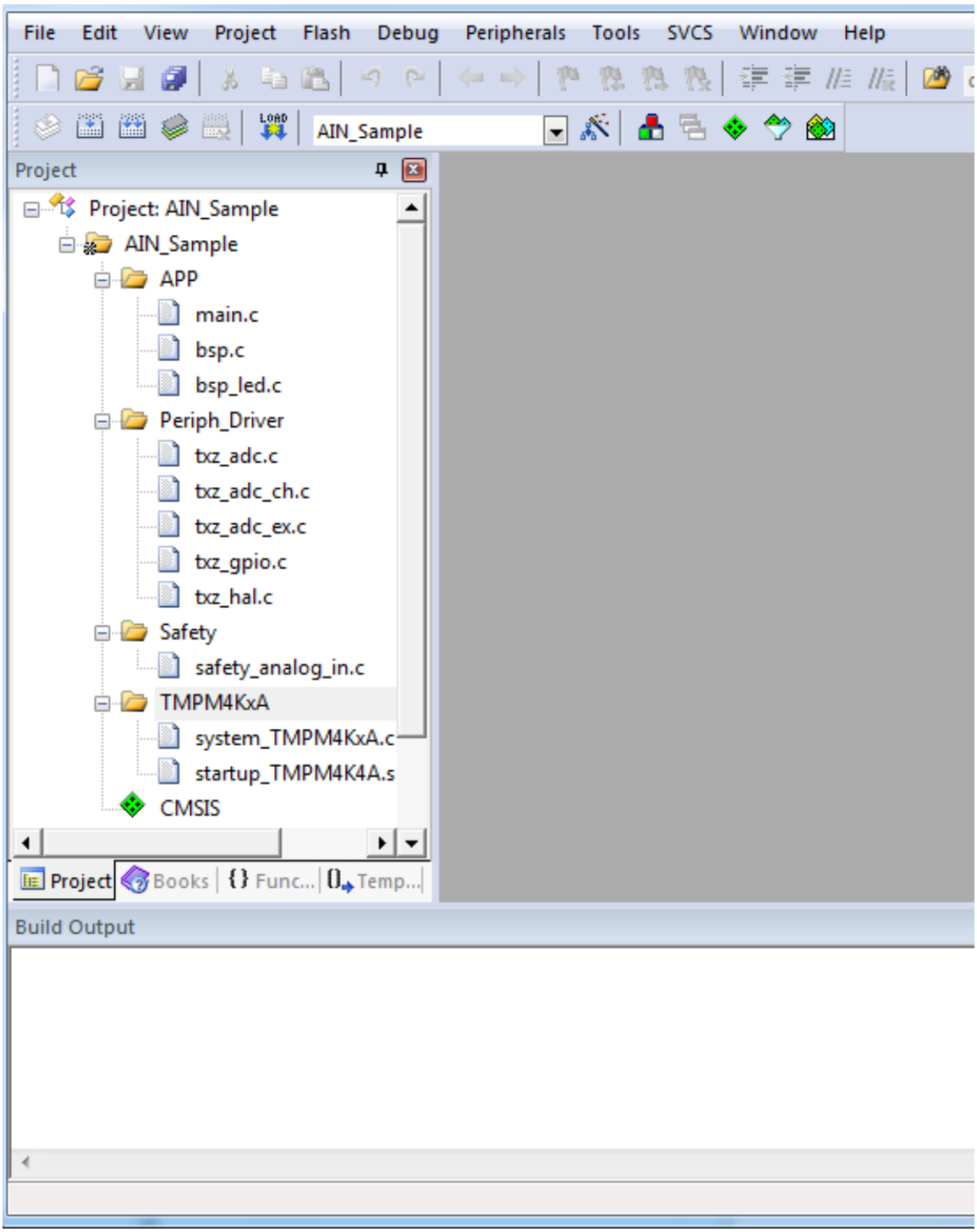
The IAR Embedded Workbench is started up.
A project can be opened in IAR Embedded Workbench by opening any workspace file (for example, AIN_Sample.eww) in the Project \ Examples \ Safety folder from the "Open Workspace" of the File menu



“Make” (F7) should be executed in the Project menu to build the project.
“Total error count: 0” shows the success of the build.

< In the case of Arm Keil >
・A project is built in the Arm Keil as follows;

The Arm Keil is started up.
A project can be opened with Arm Keil by opening any workspace file (for example, AIN_Sample.uvprojx) under the Project \ Examples \ Safety folder from "Open Project" of the Project menu



“Build Target” (F7) in the Project menu should be executed to build the project.
If the total error count is 0, the build has succeeded.

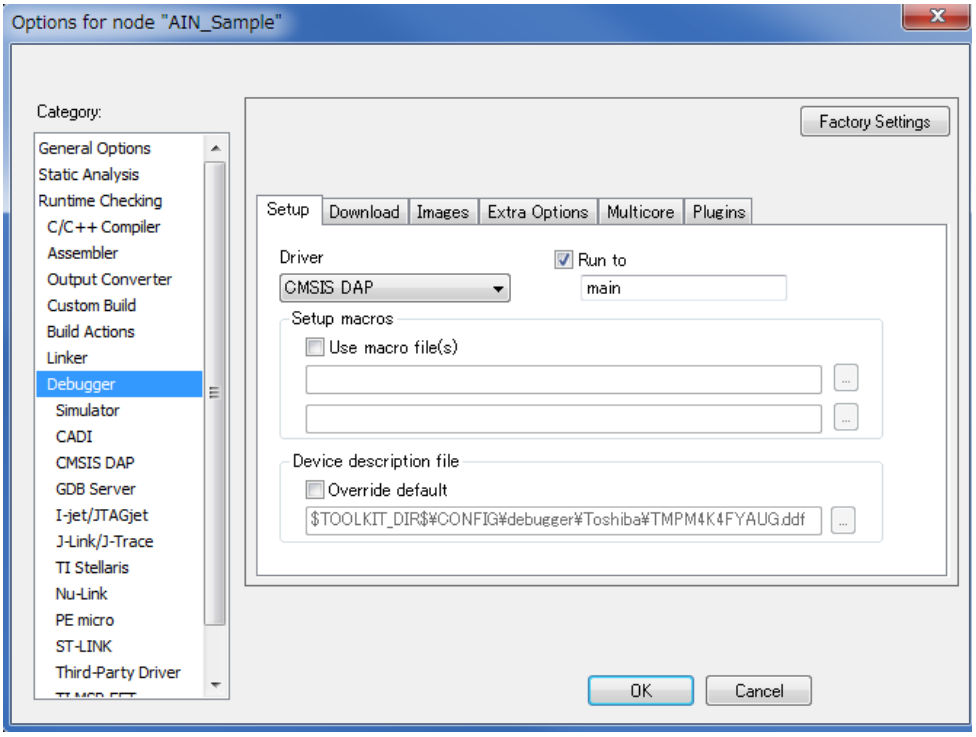
3.3. Download and Execution

The USB terminal (the DAP connector) on the evaluation board should be connected to the PC.

*Please set according to your development environment

<In the case of IAR Embedded Workbench>
The download is done in the IAR Embedded Workbench as follows;

“Option” in the Project menu in the IAR Embedded Workbench should be selected. “CMSIS DAP” should be selected in “Debugger”/“Setup”.
In “CMSIS DAP”/“Interface”, “SWD” should be selected and “OK” is clicked.

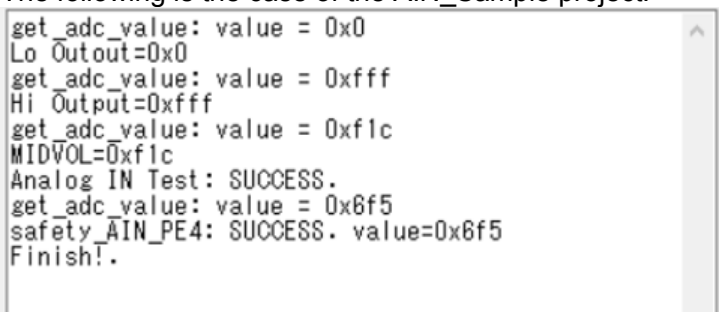


“Download and Debug” should be selected in the Project menu in the IAR Embedded Workbench, or the “Download and Debug” icon should be clicked on the tool bar. Then, the build result is written to the Flash memory in MCU, and the debugger starts up after the reset of the CPU.
There is another way. “Download” in the Project menu -> “Download of Active Application” is selected, then the build result is written to the Flash memory in MCU. After this, if the “Debug without Downloading” is selected, the debugger starts up after the reset of the CPU.

When the debugger starts up, the execution suspends at the entry of the “main” function. “Execution” (F5 key) in the Debug menu is done, the downloaded program starts. The test result is shown with the LED lighting in the self-diagnosis sample program.

The output to the terminal I/O has been disabled in the initial setting at release.
The option setting should be changed. Then the execution message can be displayed by the DBUGMSG macro in the program on the “Terminal I/O” window.
Refer to "Use of the DEBUGMSG Macro" for setting options.

The following is the case of the AIN_Sample project.

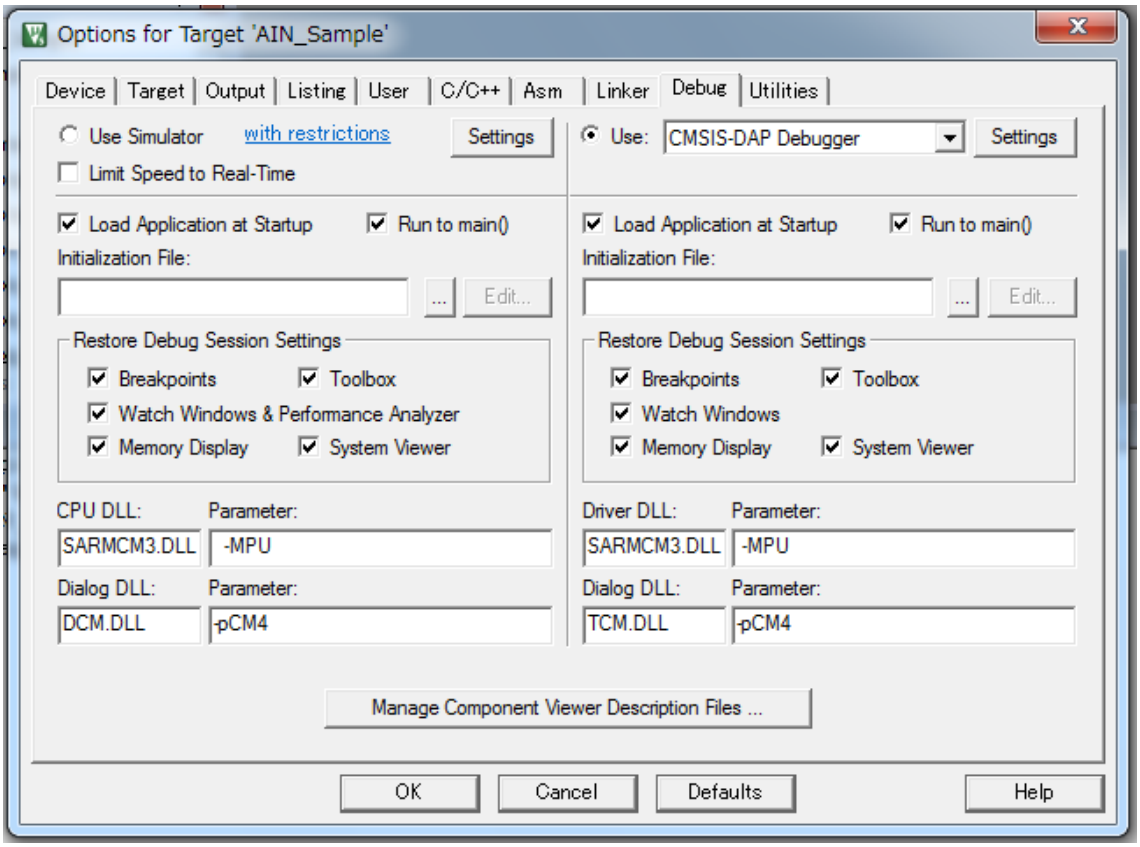


As well as the “Execution” (F5 key), “Step Over” (F10 key) and “Step In” (F11 key) are available.
“Registers” should be selected in the menu on the View to show the CPU register information.

In order to stop the debug execution, “Stop Debugging” in the Debug menu should be executed or the “x” icon (Stop Debugging) in the task bar should be selected.

< In the case of Arm Keil >
The download is done in Arm Keil as follows;

“Options for Target ‘(Project Name)’ ” should be selected in the Project menu of the Arm Keil. “CMSIS-DAP Debugger” is selected in “Debug”.

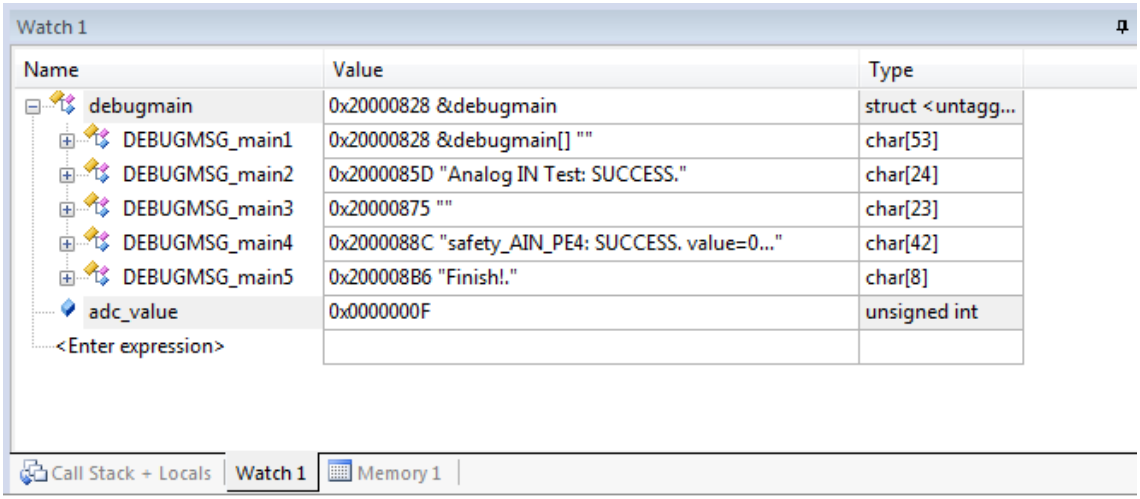


“Start/Stop Debug Session” should be selected in the Debug menu of the Arm Keil. Then, the build result is written to the Flash memory in the MCU, and the debugger starts up after the reset of the CPU.

When the debugger starts up, the execution suspends at the entry of the “main” function. “Run” (F5 key) in the Debug menu is selected, the downloaded program starts. The test result is shown with the LED lighting in the self-diagnosis sample program.

The terminal I/O cannot be used in the Arm Keil.
After the program is executed, the variable (“debugmain”) which stores the output message should be checked by the “Watch” function.
The target variable should be selected (double-click) and a right click is done. “Add ‘(Selected variable name)’ to...” should be selected. Then the “Watch” function can be done.
There is another way to check it. “Watch Windows” should be selected in the View menu. And the target variable should be dragged and dropped on the Watch window.

The following is an example of the AIN_Sample project.



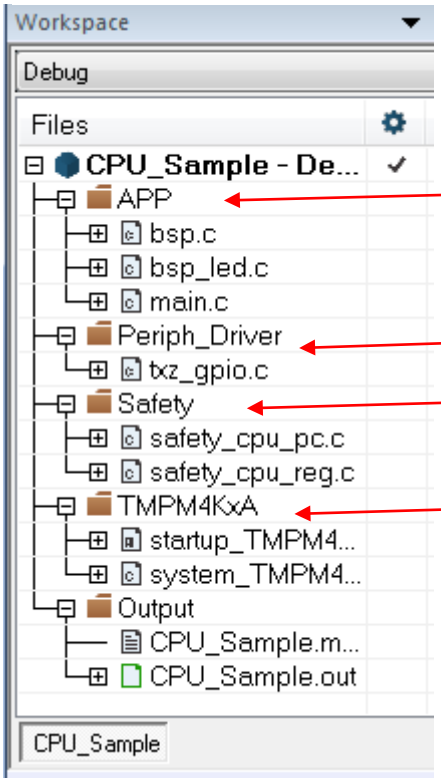
As well as “Run” (F5 key), “Step Over” (F10 key) and “Step” (F11 key) are available.
“System Viewer” in the View menu should be selected to display the CPU register information.

“Start/Stop Debug Session” in the Debug menu should be selected to stop the debug process.

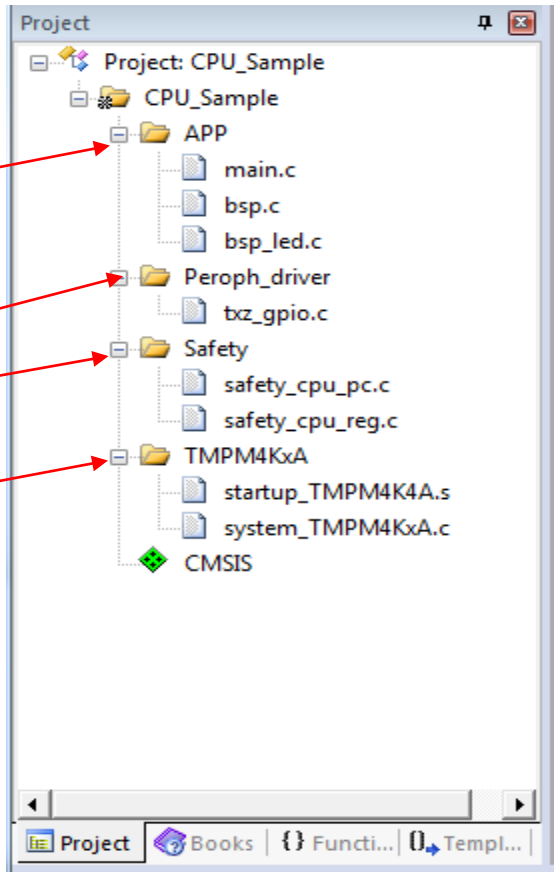
3.4. Common Description for Sample Program

The common structure of each sample project is as follows;
This is the case of “CPU_Sample” program.

IAR Embedded Workbench



Arm Keil



1
2
3
4

1: APP group icon
All files in this group are included under Project\Examples folder.
Only “main.c” among them is included in the “src” folder in the project. It has all original implementation of the sample project. The other files use the existing library codes as they are.

2: Periph_driver group icon
All files in this group are the driver files which are included in the Libraries\Periph_driver folder.
Only “txz_gpio” driver is used in the CPU sample project above.

3: Safety group icon
All files in this group are the self-diagnosis library files in the Libraries\Safety folder.

4: TMPM4KxA group icon (the “product name” should be modified to the name of the target product.)
All files in this group are the start-up codes in the Libraries\CMSIS folder.

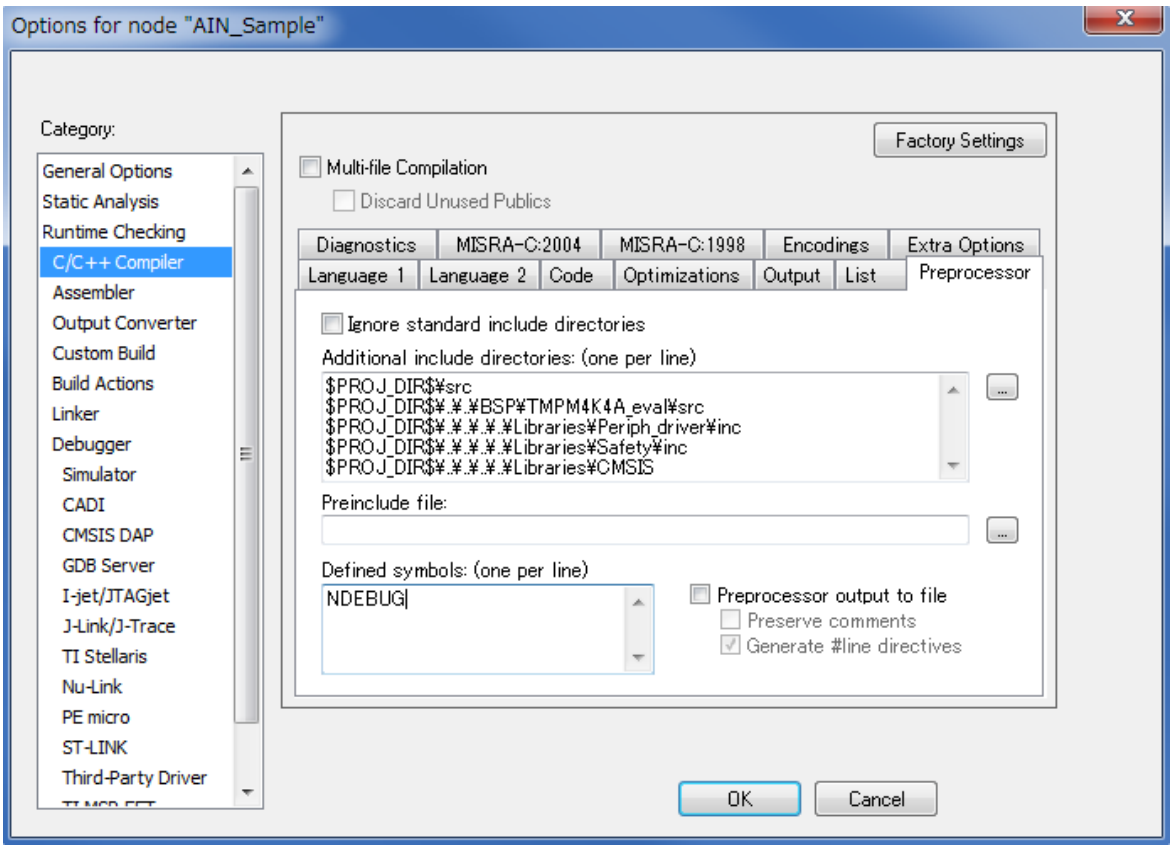
When these source codes are used, it is necessary that the path of the corresponding include file is specified.

<In the case of IAR Embedded Workbench>

The following setting should be done on the “C/C++ Compiler”/“Preprocessor” setting in the Option setting view which is opened by “Option” in the Project menu in the IAR Embedded Workbench.

Include setting

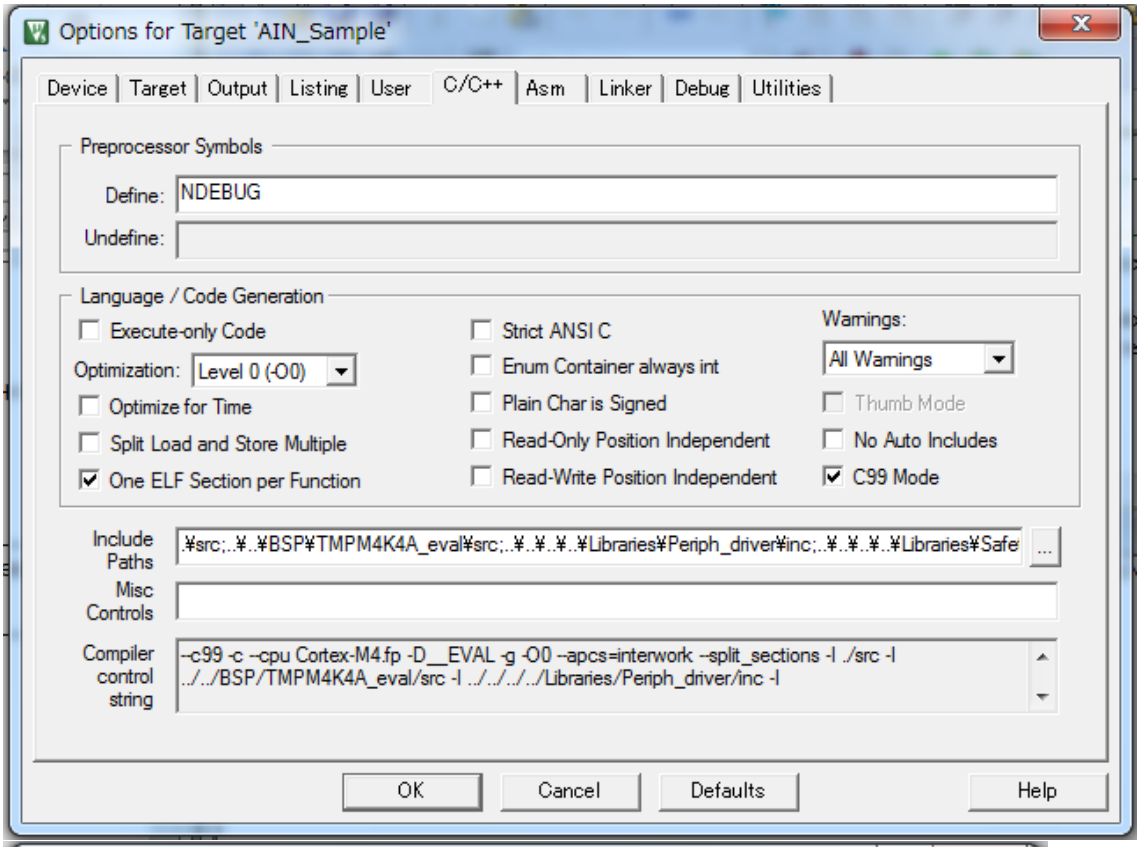
```
$PROJ_DIR$\src
$PROJ_DIR$\..\..\BSP\TMPM4K4A_eval\src
$PROJ_DIR$\..\..\..\..\Libraries\Periph_driver\inc
$PROJ_DIR$\..\..\..\..\Libraries\Safety\inc
$PROJ_DIR$\..\..\..\..\Libraries\CMSIS
```

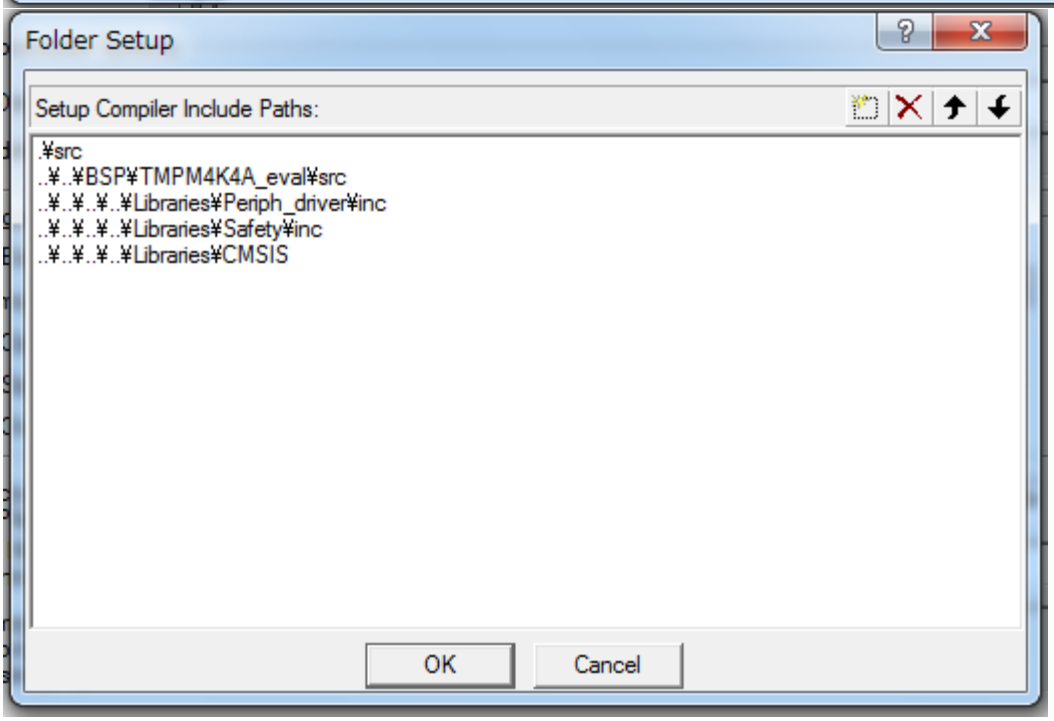


< In the case of Arm Keil >
“Include Paths” are set on the “C/C++” setting in the Option setting view which is opened by “Options for Target ‘AIN_Sample’ ” in the Project menu of the Arm Keil.

Include setting

.\src;..\..\BSP\TMPM4K4A_eval\src;..\..\..\..\Libraries\Periph_driver\inc;..\..\..\..\Libraries\Safety\inc;..\..\..\..\Libraries\CMSIS





The other setting items in this project are shown as follows.
Refer to the tables when a new project is built.

IAR Embedded Workbench

General options/Target tab	“Device” in “Processor variant” is selected. Toshiba -> TMPM4K -> TMPM4K4 -> “Toshiba TMPM4K4FYAUG” should be selected in the list. “VFPv4 single precision” should be set to “FPU” in “Floating point settings”.
General options/ Library Configuration tab	“Use CMSIS" should be checked in “CMSIS”. “Normal” should be selected for “Library”. For Library low-level interface implementation, “None” should be selected.
C/C++ compiler/Preprocessor tab	“Added include directory” should be set, if necessary. “NDEBUG” should be assigned to the symbol definition.
Linker/Checksum tab	The CRC32 calculation range of the Flash memory should be set when the safety_FLASH is used. (Refer to the FLASH test item in the Memory Self-diagnosis Program Application Note)
Debugger/Setting tab	“CMSIS DAP” should be selected as a driver.
J-Link/J-Trace/Connection	“SWD” should be selected for “Interface”.

Arm Keil

Option for Target/Device tab	“TMPM4K4FYAUG” should be selected for “Device”.
Manage Run-Time Environment icon	“CORE” in “CMSIS” should be checked.
Option for Target/C/C++ tab	“Added include directory” should be set, if necessary. “NDEBUG” should be assigned to the symbol definition.
Option for Target/Debug tab	“CMSIS-DAP Debugger” should be selected for “Use:”.

3.5. Use of DEBUGMSG Macro

The self-diagnosis library and the sample program can output messages to the development environment using the DEBUGMSG macro.
The DEBUGMSG macro can be used only in the IAR Embedded Workbench, not in the Arm Keil. As for the Arm Keil, for the details of the message check, refer to the download in the case of Arm Keil in “Download and Execution”.

<In the case of IAR Embedded Workbench>

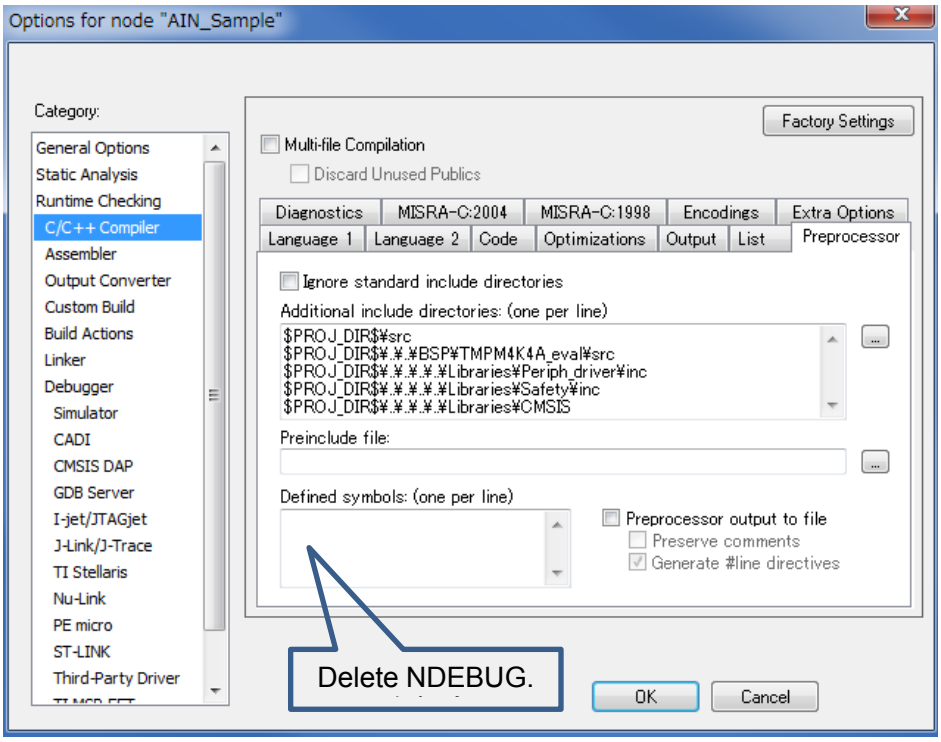
```
DEBUGMSG(1, ("jump to function address 0x%08x SUCCESS.\r\n", (uint32_t)func1));
```

This macro is defined in the following file.

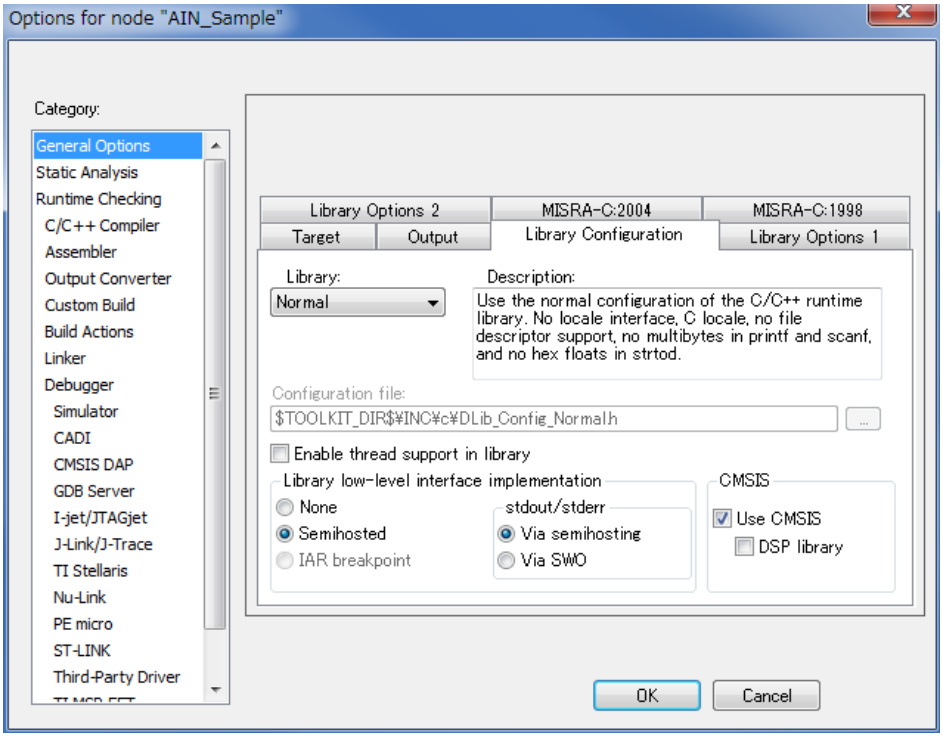
From Libraries\Safety\inc\txz_safety_def.h

```
// DEBUG define
#ifndef NDEBUG
#include <stdio.h>
// message output in DEBUG build
// use DEBUGMSG(1, ("message %s", string)) kind of forms to pass multiple arguments
// #define DEBUGMSG(x, y)          (((x)?((void)printf y),1:0))
#define DEBUGMSG(x, y)       if(x){((void)printf y);}
#else
// if you would like debug in "RELEASE" build, enable following and make build option
// general - library option - low level library option as semi-hosting
//#include <stdio.h>
//#define DEBUGMSG(x, y)          ((void)((x)?(printf y),1:0))
#define DEBUGMSG(x, y)            ((void)0)
#endif
```

When NDEBUG has been defined in the project, the DEBUGMSG macro does nothing. When NDEBUG is not defined, the macro is expanded to the formula which calls “printf”.



In “General option”/“Library setting” in the Project setting, “Semihosting” and “Via Semihosting” are specified for the implementation of the low level interface library. Then, the “printf” output sentence described in the DEBUGMSG macro is displayed on the terminal I/O window in the IAR Embedded Workbench environment.



On the other hand, the TMPM4KxA sample program has the bsp_uart_io module (Project\Examples\BSP\TMPM4K4A_eval\src\bsp_uart_io.c) which outputs the printf sentence to the UART. It is notified that the module may compete with the console I/O setting in the IAR Embedded Workbench environment.

4. Revision History

Revision	Date	Description
1.0	2019-08-27	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.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- 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.**