

自己診断プログラム アプリケーションノート 基本設定

概要

本アプリケーションノートは、自己診断プログラムを使用する際の基本設定の資料です。
TXZ シリーズマイコンを使用して自己診断プログラムを使用するときに参考としてください。
TXZ シリーズのペリフェラルドライバが同梱されたサンプルプログラムに対応しています。

目次

概要	1
目次	2
1. はじめに.....	4
2. 動作確認環境.....	5
3. 設定方法.....	6
3.1. ツリー構成	6
3.2. ビルド方法	7
3.3. ダウンロードと実行.....	9
3.4. サンプルプログラム共通の説明	14
3.5. DEBUGMSG マクロの使用	18
4. 改訂履歴.....	20
製品取り扱い上のお願い.....	21

Arm、Cortex および Keil は、Arm Limited（またはその子会社）の米国およびその他の国における登録商標です。

この資料に記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

1. はじめに

本アプリケーションノートは、自己診断プログラムを使用する際の基本設定の資料です。
各テストを実施する際の設定方法や動作環境などの情報を記載しています。
本ライブラリコードは、公開中のサンプルプログラムに追加(上書き)して使用します。

本資料の説明は、TMPM4K グループ(1)をベースに記載しています。
各製品への展開は、製品仕様に合わせて変更をお願いします。

2. 動作確認環境

使用マイコン
統合開発環境
統合開発環境

TMPM4K4AFYAUG
IAR Embedded Workbench for ARM 8.22.2
Arm® Keil® MDK Version 5.24.2.0

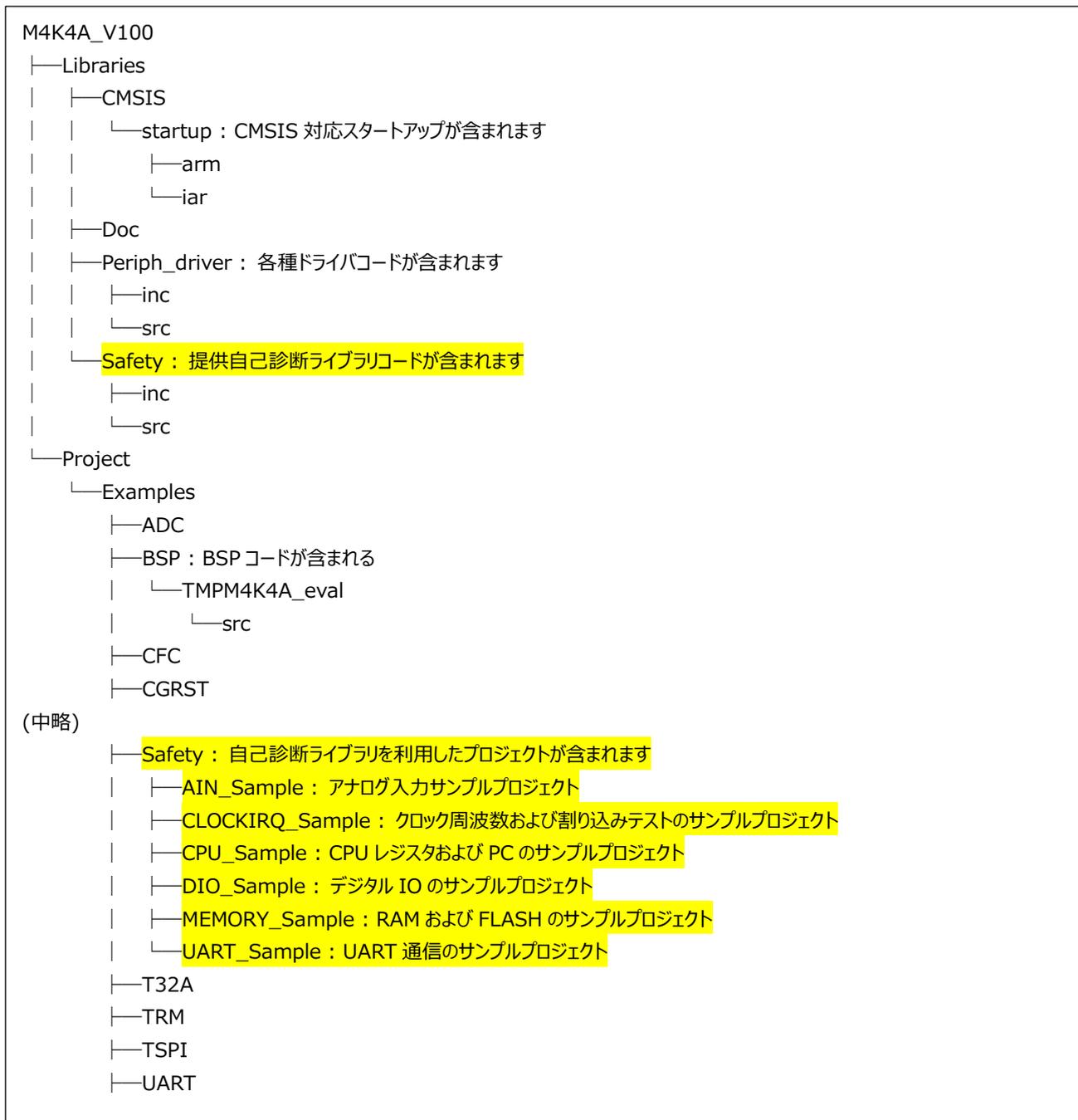
3. 設定方法

TMPM4KxA のサンプルプログラムを使用した場合の使用例です。

本ライブラリコードは、公開中の TMPM4KxA サンプルプログラムのバージョン v1.0.0 に追加(上書き)して使用します。

3.1. ツリー構成

サンプルプログラム v1.0.0 の上に提供ツリーを上書き展開すると以下のようなフォルダツリー形式となります。



自己診断ライブラリの各関数は必ずいずれかのサンプルプロジェクトに含まれるようになっており、サンプルプロジェクトのいずれかをビルドして TMPM4K4 評価ボード にダウンロード実行することで動作を確認することができます。

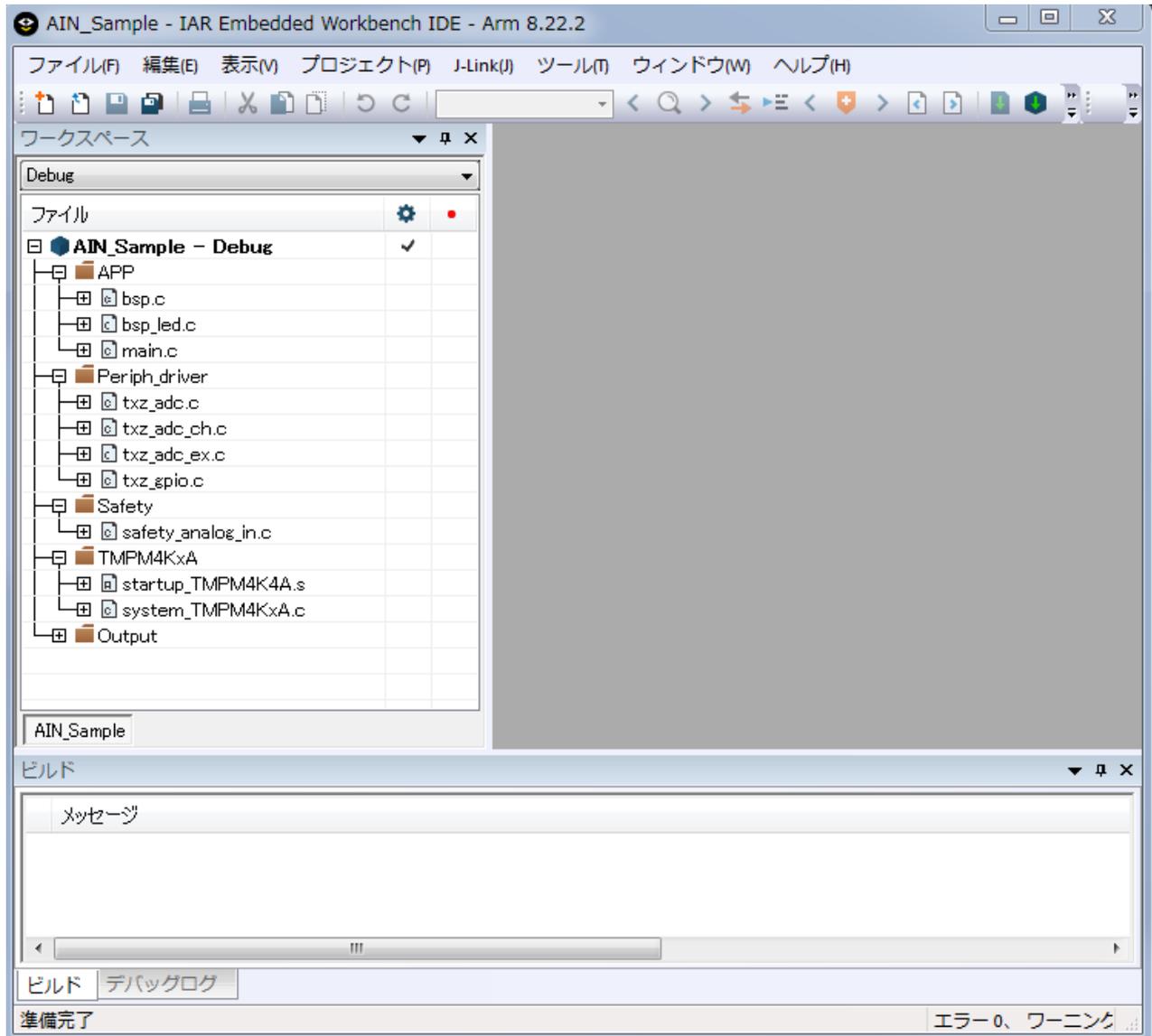
色つき部分が自己診断用の追加ファイルです。

3.2. ビルド方法

<IAR Embedded Workbench の場合>

- ・ IAR Embedded Workbench でのビルド方法は以下のとおりとなります。

IAR Embedded Workbench を起動し、ファイルメニューの「ワークスペースを開く」より、Project¥Examples¥Safety フォルダ以下の任意のワークスペースファイル(例えば、AIN_Sample.eww)を開くと、IAR Embedded Workbench で プロジェクトを開くことができます。

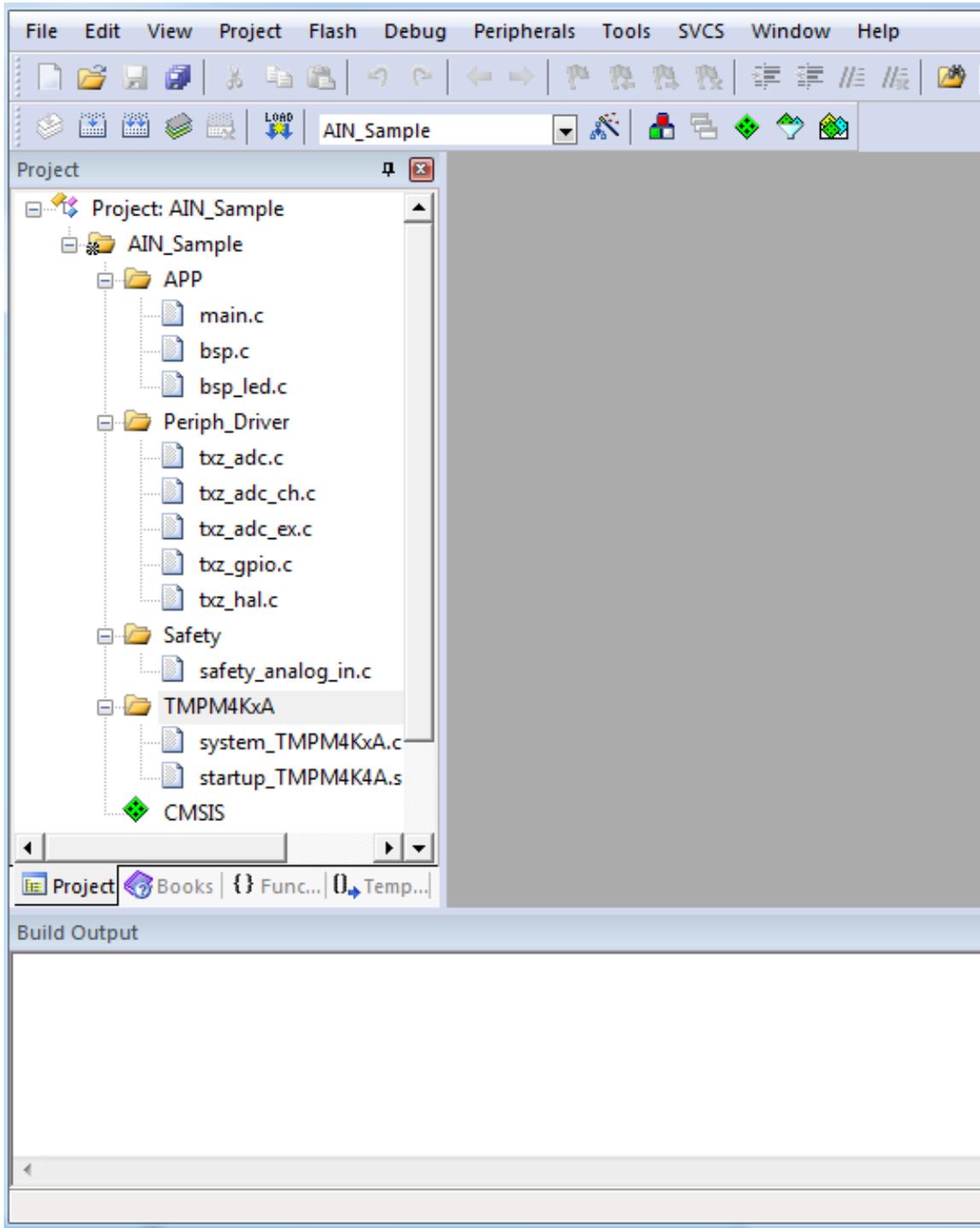


プロジェクトメニューの「メイク」(F7) を実行することにより、ビルドを行います。
エラーの合計数: 0 であればビルド成功です。

< Arm Keil の場合 >

- Arm Keil でのビルド方法は以下のとおりとなります。

Arm Keil を起動し、Project メニューの「Open Project」より、Project¥Examples¥Safety フォルダ以下の任意のワークスペースファイル(例えば、AIN_Sample.uvprojx)を開くと、Arm Keil でプロジェクトを開くことができます。



Project メニューの「Build Target」(F7) を実行することにより、ビルドを行います。
エラーの合計数: 0 であればビルド成功です。

3.3. ダウンロードと実行

評価ボードの USB 端子 (DAP コネクタ) を PC に接続します。

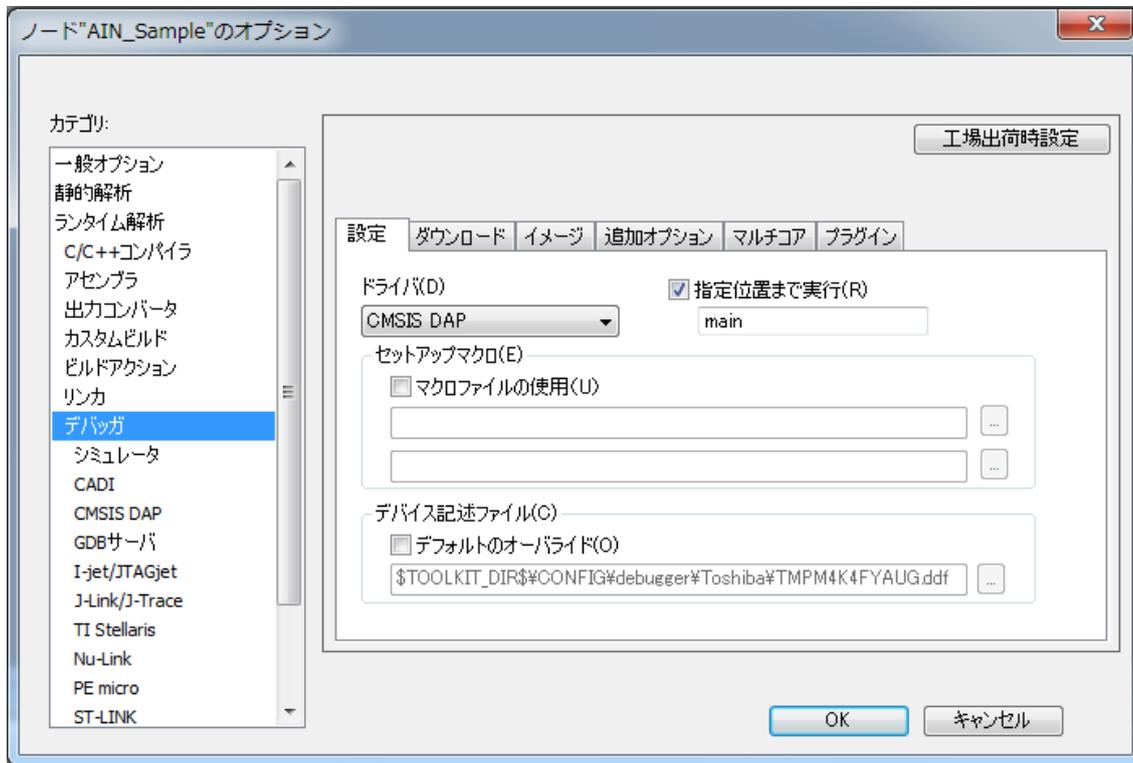
※ご使用の開発環境に合わせて設定を実施してください。

<IAR Embedded Workbench の場合>

- ・ IAR Embedded Workbench でのダウンロード方法は以下のとおりとなります。

IAR Embedded Workbench のプロジェクトメニュー、「オプション」を選択、デバッガ/設定で「CMSIS DAP」を選択します。

CMSIS DAP/接続で「SWD」を選択して OK をクリック。



この状態で、IAR Embedded Workbench のプロジェクトメニューから「ダウンロードしてデバッグ」を選択するか、ツールバー上の「ダウンロードしてデバッグ」アイコンをクリックすると、ビルド結果を評価ボード上の FLASH メモリに書き込み、CPU リセット後デバッガが起動します。

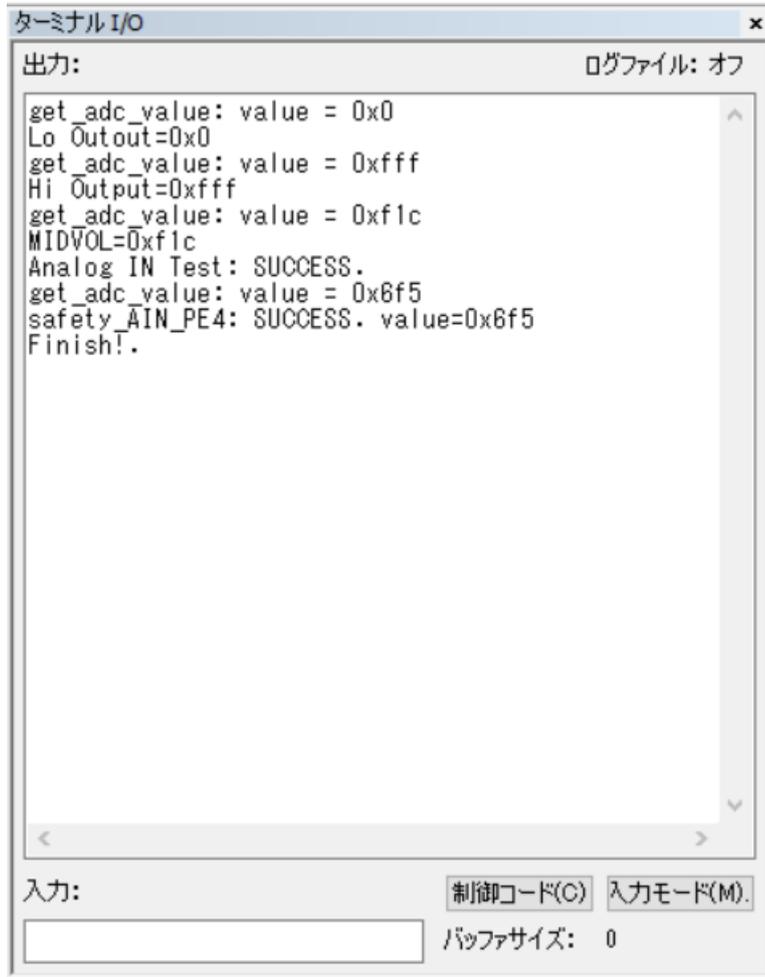
別の方法として、プロジェクトメニューから「ダウンロード」 → 「アクティブなアプリケーションのダウンロード」を選択すると、ビルド結果を評価ボード上の FLASH メモリに書き込みます。このあとプロジェクトメニュー「ダウンロードしないでデバッグ」を選択すると、CPU リセット後デバッガが起動します。

デバッガを起動すると、main 関数のエントリで実行が停止します。このあとデバッグメニューから「実行」(F5 キー)を実行すると、ダウンロードしたプログラムが実行されます。自己診断サンプルプログラムでは、テスト結果により LED が点灯します。

リリース時の設定では、ターミナル I/O への出力は無効になっています。

オプション設定を切り替えることで、プログラム中の DEBUGMSG マクロで、「ターミナル I/O」ウィンドウに実行中のメッセージを出力することができます。オプションの設定は、「DEBUGMSG マクロの使用」を参照してください。

AIN_Sample プロジェクト例の場合は以下になります。



The screenshot shows a window titled "ターミナル I/O" (Terminal I/O). The window has a title bar with a close button (x) on the right. Inside the window, there is a text area with the following output:

```
出力:                                ログファイル: オフ
get_adc_value: value = 0x0
Lo Output=0x0
get_adc_value: value = 0xffff
Hi Output=0xffff
get_adc_value: value = 0xf1c
MIDVOL=0xf1c
Analog IN Test: SUCCESS.
get_adc_value: value = 0x6f5
safety_AIN_PE4: SUCCESS. value=0x6f5
Finish!.
```

At the bottom of the window, there is an input field labeled "入力:" (Input:). To the right of the input field are two buttons: "制御コード(C)" (Control Code (C)) and "入力モード(M)" (Input Mode (M)). Below the input field, there is a label "バッファサイズ: 0" (Buffer Size: 0).

また、「実行」(F5 キー)以外にも「ステップオーバ」(F10 キー)や「ステップイン」(F11 キー)を使用可能です。

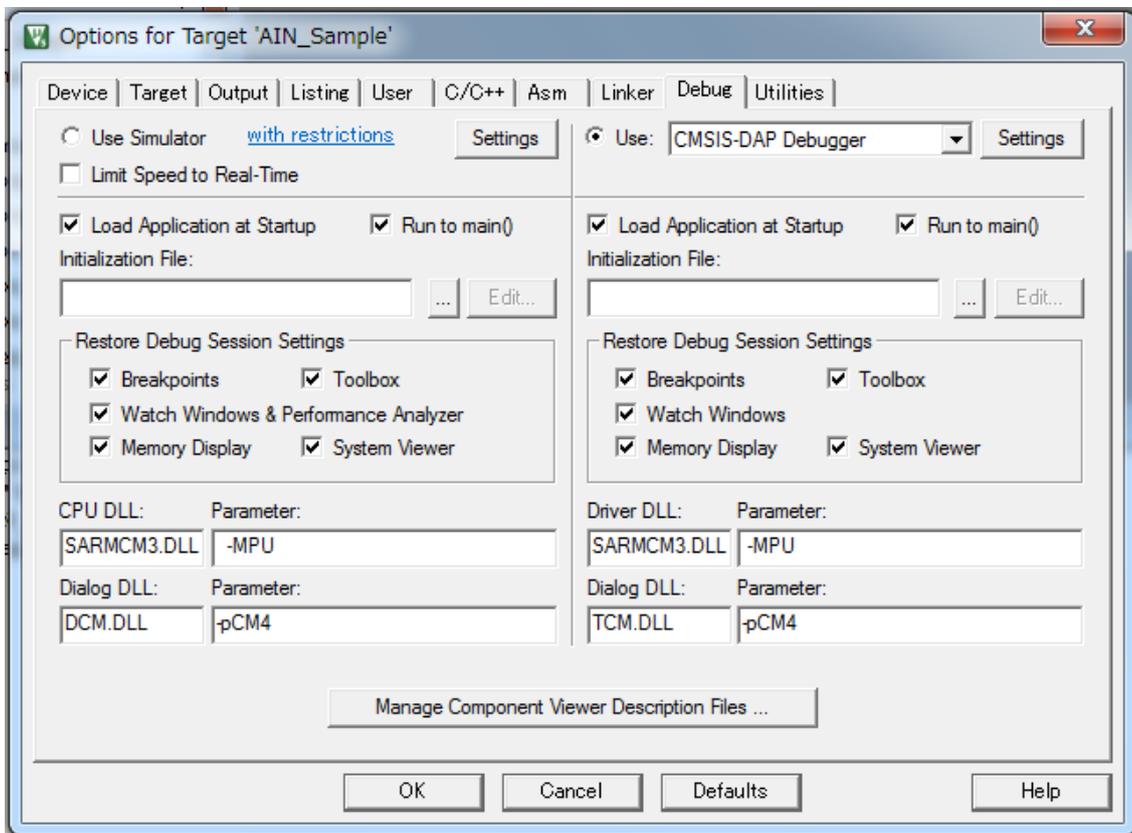
CPU レジスタ表示を見たいときは表示メニューから「レジスタ」を選択して表示させてください。

デバッグ実行を中止するには、デバッグメニューから「デバッグの停止」を実行するか、タスクバーの「x」アイコン(デバッグの中止)を実行してください。

< Arm Keil の場合 >

- ・ Arm Keil でのダウンロード方法は以下のとおりとなります。

Arm Keil の Project メニュー、「Options for Target '(Project Name)」」を選択、DEBUG で「CMSIS-DAP Debugger」を選択します。



この状態で、Arm Keil の Debug メニューから「Start/Stop Debug Session」を選択すると、ビルド結果を評価ボード上の FLASH メモリに書き込み、CPU リセット後デバッガが起動します。

デバッガを起動すると、main 関数のエントリで実行が停止します。このあと Debug メニューから「Run」(F5 キー) を実行すると、ダウンロードしたプログラムが実行されます。自己診断サンプルでは、テスト結果により LED が点灯します。

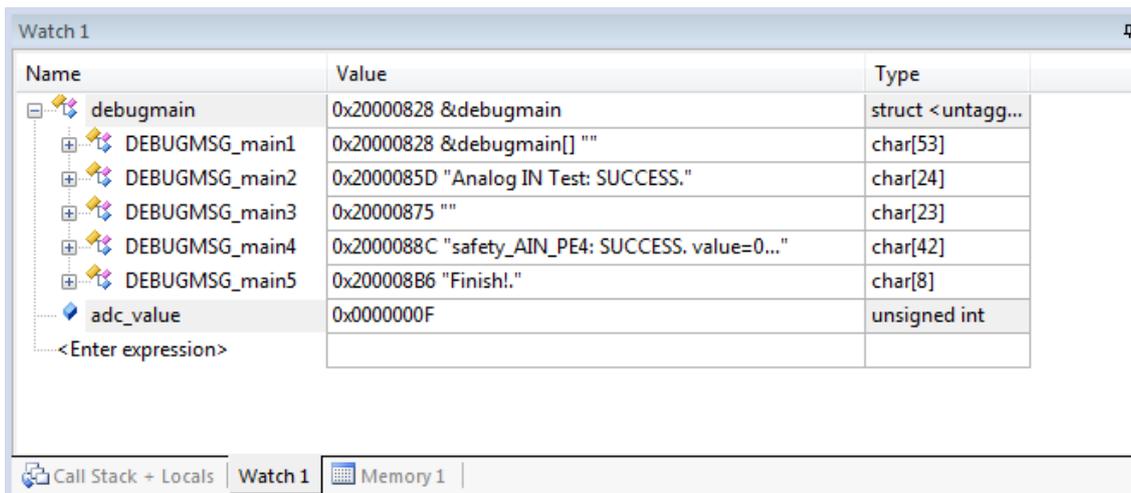
Arm Keil では、ターミナル I/O の使用が不可能となります。

プログラム実行後、Watch 機能を使用して出力メッセージが格納された変数(debugmain)を確認します。

Watch 機能は、確認したい変数を選択(ダブルクリック)し、右クリックから「Add '(選択した変数名)' to...」を選択すると、Watch することが可能です。

別の方法として、View メニューから「Watch Windows」を選択し、確認したい変数を Watch ウィンドウにドラッグ/ドロップすることで確認できます。

AIN_Sample プロジェクト例の場合は以下になります。



The screenshot shows a debugger's Watch window titled "Watch 1". It contains a table with three columns: Name, Value, and Type. The table lists several variables, including a struct pointer and several character arrays containing debug messages. At the bottom, there are tabs for "Call Stack + Locals", "Watch 1", and "Memory 1".

Name	Value	Type
debugmain	0x20000828 &debugmain	struct <untagg...
DEBUGMSG_main1	0x20000828 &debugmain[] ""	char[53]
DEBUGMSG_main2	0x2000085D "Analog IN Test: SUCCESS."	char[24]
DEBUGMSG_main3	0x20000875 ""	char[23]
DEBUGMSG_main4	0x2000088C "safety_AIN_PE4: SUCCESS. value=0..."	char[42]
DEBUGMSG_main5	0x200008B6 "Finish!"	char[8]
adc_value	0x0000000F	unsigned int
<Enter expression>		

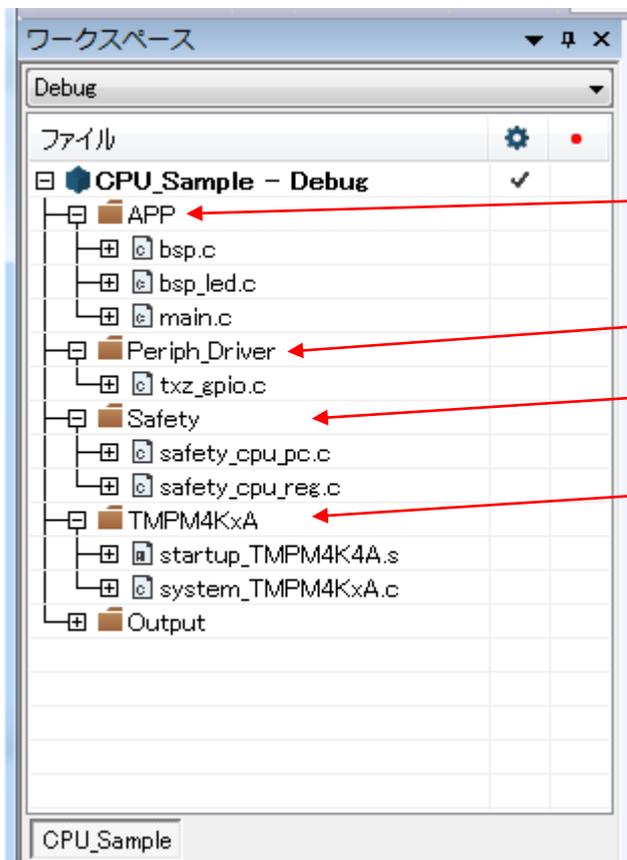
また、「Run」(F5 キー)以外にも「Step Over」(F10 キー)や「Step」(F11 キー)を使用可能です。CPU レジスタ表示を見たいときは View メニューから「System Viewer」を選択して表示させてください。

デバッグ実行を中止するには、Debug メニューから「Start/Stop Debug Session」を実行してください。

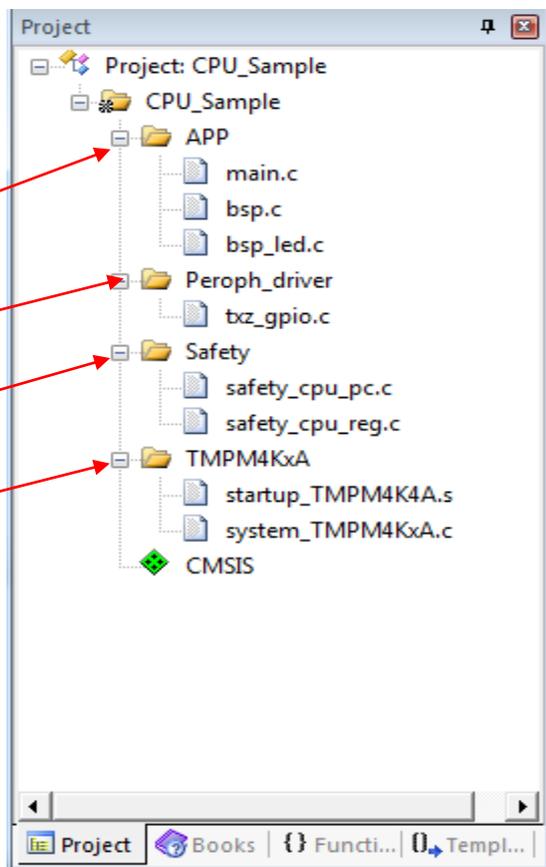
3.4. サンプルプログラム共通の説明

各サンプルプロジェクトは、以下のような共通の構成となっています。
以下は CPU_Sample の場合です。

IAR Embedded Workbench



Arm Keil



1: APP グループアイコン

このグループ内のファイルは全て Project¥Examples 以下に含まれます。
このうち main.c のみがこのプロジェクトの src フォルダ内に含まれ、サンプルプロジェクト独自の実装を全て含んでいます。これ以外のファイルは既存のライブラリコードをそのまま利用しています。

2: Periph_driver グループアイコン

このグループ内のファイルは全て Libraries¥Periph_driver フォルダ以下に含まれているドライバファイルです。上記例の CPU サンプルプロジェクトでは txz_gpio ドライバのみ使用しています。

3: Safety グループアイコン

このグループ内のファイルは全て Libraries¥Safety フォルダ以下に含まれている自己診断ライブラリファイルです。

4: TPM4KxA グループアイコン (製品により製品名が変更になります)

このグループ内のファイルは全て Libraries¥CMSIS フォルダ以下に含まれているスタートアップコードです。

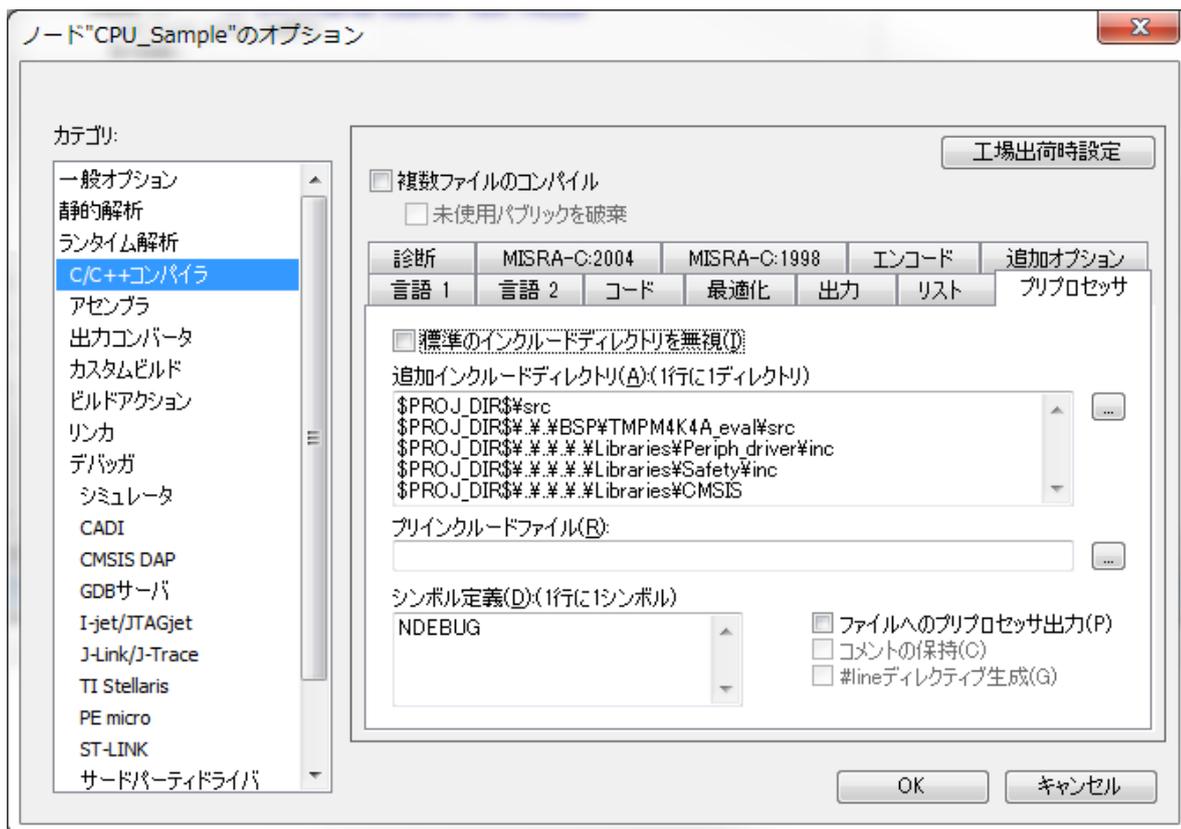
これらのソースコードを利用するには、対応するインクルードファイルのパスを指定する必要があります。

<IAR Embedded Workbench の場合>

IAR Embedded Workbench のプロジェクトメニュー、「オプション」で開くオプション設定画面の C/C++コンパイラ/プリプロセッサ設定で、以下のように設定します。

インクルードの設定

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



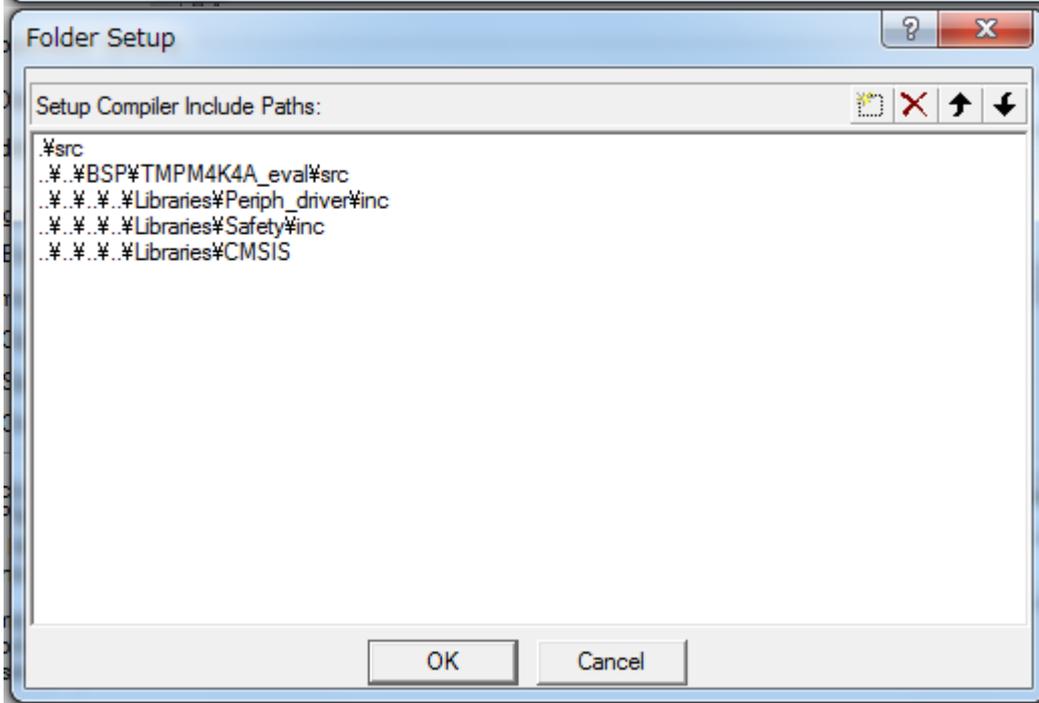
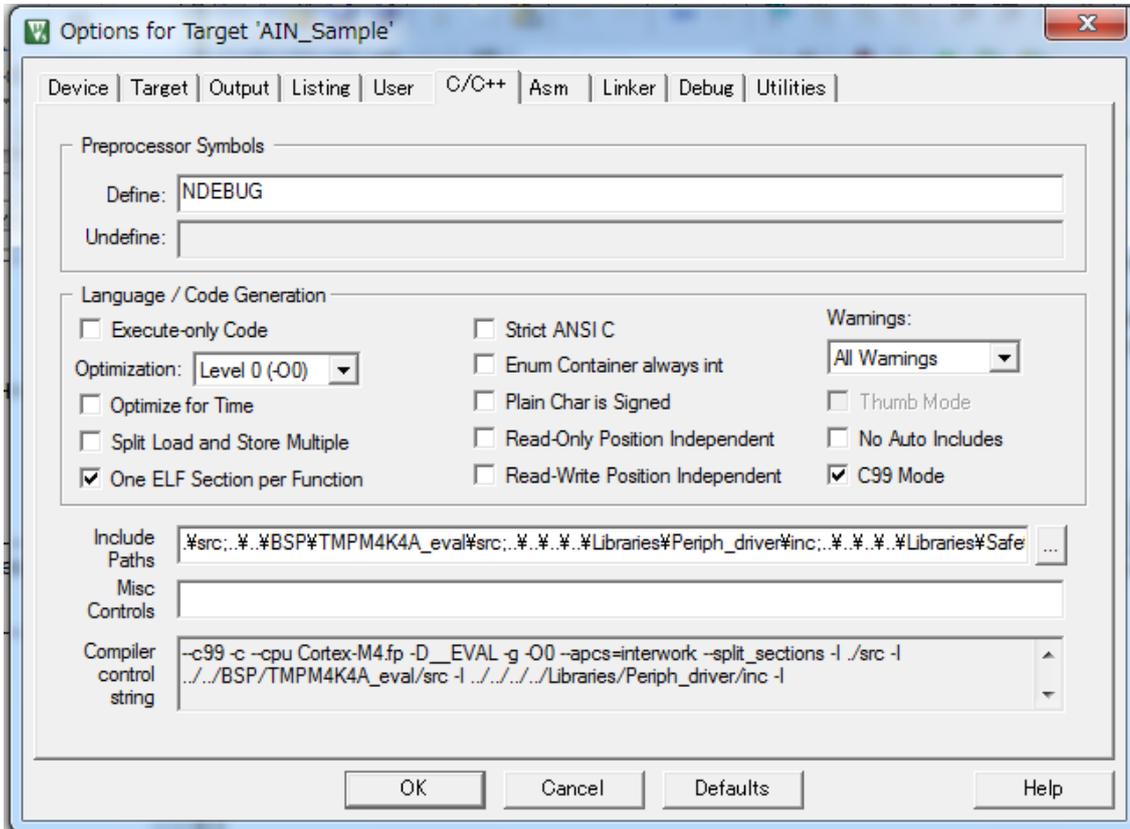
< Arm Keil の場合 >

Arm Keil では Project メニュー、「Options for Target 'AIN_Sample」で開くオプション設定画面の C/C++ 設定で、Include Paths を以下のように設定します。

インクルードの設定

```

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



このほか、プロジェクトで設定している項目には以下があります。新規にプロジェクトを作成する際に参考にしてください。

IAR Embedded Workbench

一般オプション/ターゲットタブ	「プロセッサ選択」で「デバイス」を選択し、リストから Toshiba → TMPM4K → TMPM4K4 → 「Toshiba TMPM4K4FYAUG」を選択 「浮動小数点演算の設定」で「FPU」に「VFPv4 single precision」を選択
一般オプション/ライブラリタブ	CMSIS で「CMSIS を利用する」にチェックを入れる 「ライブラリ」に「ノーマル」を選択 低レベルインタフェースのライブラリ実装は「なし」を選択
C/C++コンパイラ/プリプロセッサタブ	必要に応じて「追加インクルードディレクトリ」を設定 シンボル定義に「NDEBUG」を指定
リンカ/チェックサムタブ	safety_FLASH を利用するときに、FLASH 領域の CRC32 計算範囲を設定する。(「メモリ用自己診断プログラムアプリケーションノート」の FLASH テスト項目を参照)
デバッガ/設定タブ	ドライバに「CMSIS DAP」を選択
J-Link/J-Trace/接続	「インタフェース」に「SWD」を選択

Arm Keil

Option for Target/Device タブ	Device に「TMPM4K4FYAUG」を選択
Manage Run-Time Environment アイコン	CMSIS の CORE にチェックを入れる
Option for Target/C/C++タブ	必要に応じて「追加インクルードディレクトリ」を設定 シンボル定義に「NDEBUG」を指定
Option for Target/Debug タブ	Use: に「CMSIS-DAP Debugger」を選択

3.5. DEBUGMSG マクロの使用

自己診断ライブラリおよびサンプルプログラムでは DEBUGMSG マクロを利用して、開発環境へのメッセージ出力が可能です。

DEBUGMSG マクロの使用は IAR Embedded Workbench 版のみ使用可能となります。Arm Keil 版でのメッセージの確認は「ダウンロードと実行」の Arm Keil でのダウンロード方法を参照してください。

<IAR Embedded Workbench の場合>

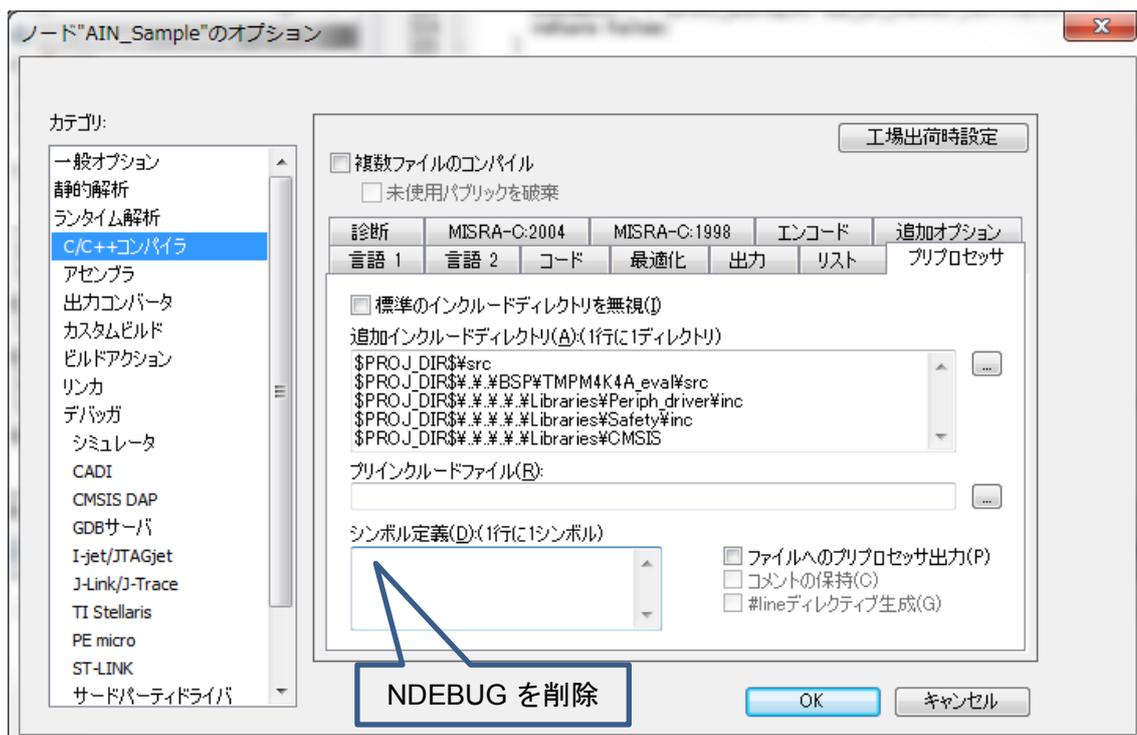
```
DEBUGMSG(1, ("jump to function address 0x%08x SUCCESS.¥r¥n", (uint32_t)func1));
```

このマクロは以下のファイルで定義されています。

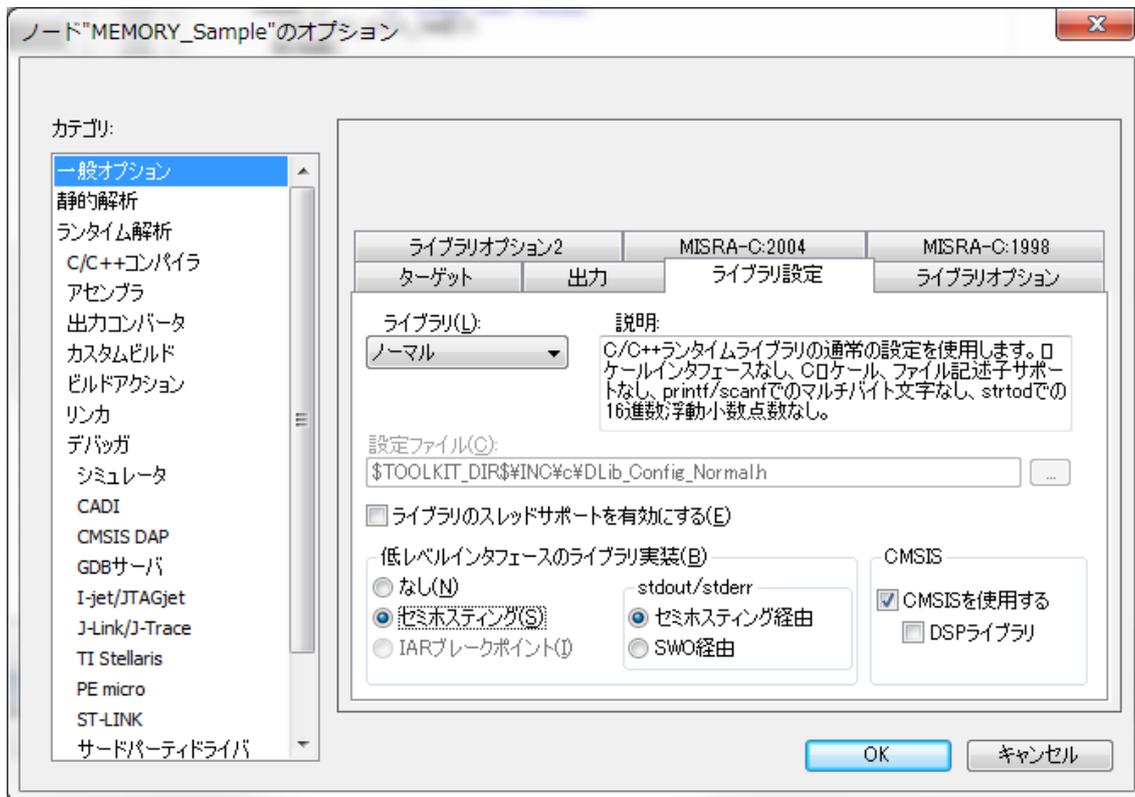
Libraries¥Safety¥inc¥txz_safety_def.h より

```
// 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
```

プロジェクトで NDEBUG が定義されていると、DEBUGMSG マクロは何もしません。NDEBUG の定義を無くした場合、printf を呼び出す式に展開されます。



また、プロジェクト設定で、一般オプション/ライブラリ設定で低レベルインタフェースのライブラリ実装に「セミホスティング」および「セミホスティング経由」を指定することにより、DEBUGMSG マクロで記述した printf 文の出力が IAR Embedded Workbench 環境のターミナル I/O ウィンドウに出力されます。



一方、TPM4KxA サンプルプログラムには、bsp_uart_io モジュール (Project¥Examples¥BSP¥TPM4K4A_eval¥src¥bsp_uart_io.c)があり、printf 文を UART に出力する実装をしています。IAR Embedded Workbench 環境のコンソール I/O 設定と競合する関係にありますのでご注意ください。

4. 改訂履歴

Revision	Date	Description
1.0	2019-08-29	初版

製品取り扱い上のお願い

株式会社東芝およびその子会社ならびに関係会社を以下「当社」といいます。

本資料に掲載されているハードウェア、ソフトウェアおよびシステムを以下「本製品」といいます。

- 本製品に関する情報等、本資料の掲載内容は、技術の進歩などにより予告なしに変更されることがあります。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。また、文書による当社の事前の承諾を得て本資料を転載複製する場合でも、記載内容に一切変更を加えたり、削除したりしないでください。
- 当社は品質、信頼性の向上に努めていますが、半導体・ストレージ製品は一般に誤作動または故障する場合があります。本製品をご使用頂く場合は、本製品の誤作動や故障により生命・身体・財産が侵害されることのないように、お客様の責任において、お客様のハードウェア・ソフトウェア・システムに必要な安全設計を行うことをお願いします。なお、設計および使用に際しては、本製品に関する最新の情報（本資料、仕様書、データシート、アプリケーションノート、半導体信頼性ハンドブックなど）および本製品が使用される機器の取扱説明書、操作説明書などをご確認の上、これに従ってください。また、上記資料などに記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価し、お客様の責任において適用可否を判断してください。
- 本製品は、特別に高い品質・信頼性が要求され、またはその故障や誤作動が生命・身体に危害を及ぼす恐れ、膨大な財産損害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある機器（以下“特定用途”という）に使用されることは意図されていませんし、保証もされていません。特定用途には原子力関連機器、航空・宇宙機器、医療機器（ヘルスケア除く）、車載・輸送機器、列車・船舶機器、交通信号機器、燃焼・爆発制御機器、各種安全関連機器、昇降機器、発電関連機器などが含まれますが、本資料に個別に記載する用途は除きます。特定用途に使用された場合には、当社は一切の責任を負いません。なお、詳細は当社営業窓口まで、または当社 Web サイトのお問い合わせフォームからお問い合わせください。
- 本製品を分解、解析、リバースエンジニアリング、改造、改変、翻案、複製等しないでください。
- 本製品を、国内外の法令、規則及び命令により、製造、使用、販売を禁止されている製品に使用することはできません。
- 本資料に掲載してある技術情報は、製品の代表的動作・応用を説明するためのもので、その使用に際して当社及び第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。
- 別途、書面による契約またはお客様と当社が合意した仕様書がない限り、当社は、本製品および技術情報に関して、明示的にも黙示的にも一切の保証（機能動作の保証、商品性の保証、特定目的への合致の保証、情報の正確性の保証、第三者の権利の非侵害保証を含むがこれに限らない。）をしておりません。
- 本製品、または本資料に掲載されている技術情報を、大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」、「米国輸出管理規則」等、適用ある輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
- 本製品の RoHS 適合性など、詳細につきましては製品個別に必ず当社営業窓口までお問い合わせください。本製品のご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用ある環境関連法令を十分調査の上、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は一切の責任を負いかねます。