Rev 1.40.00

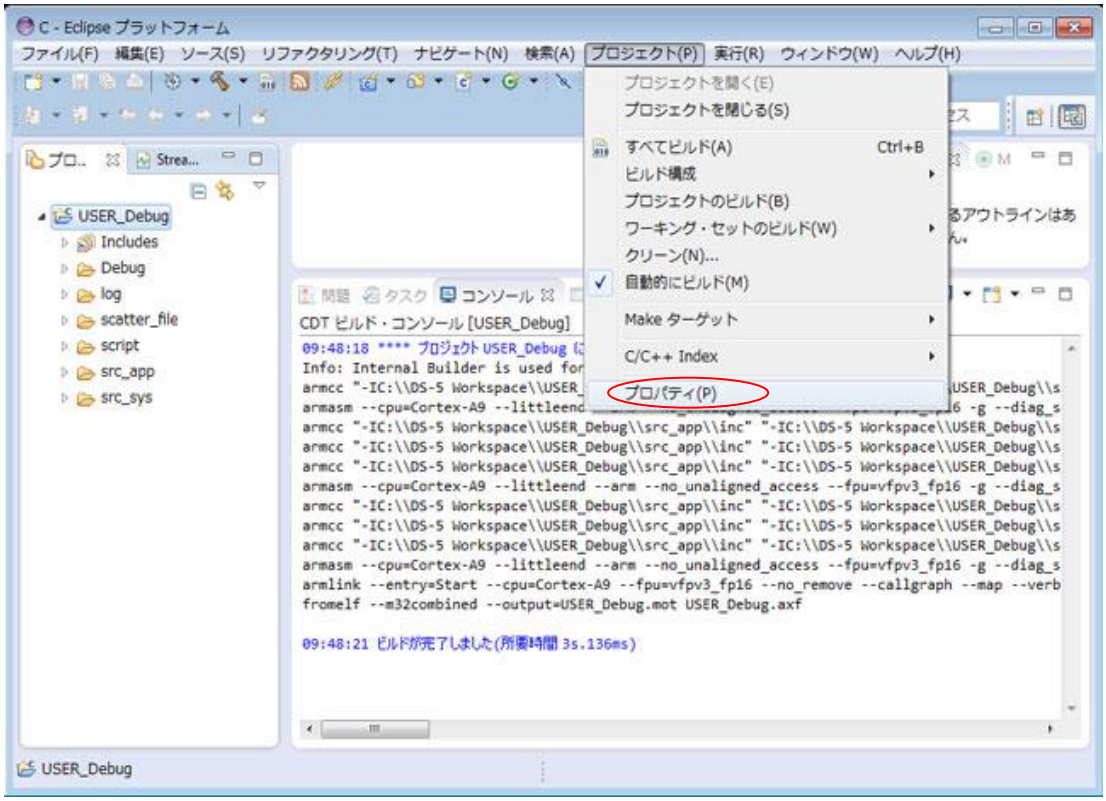
DS-5（ARMC）ツールチェインの設定と必要事項の説明

（ルネサス　ＲＺ／Ａ１Ｈ用）

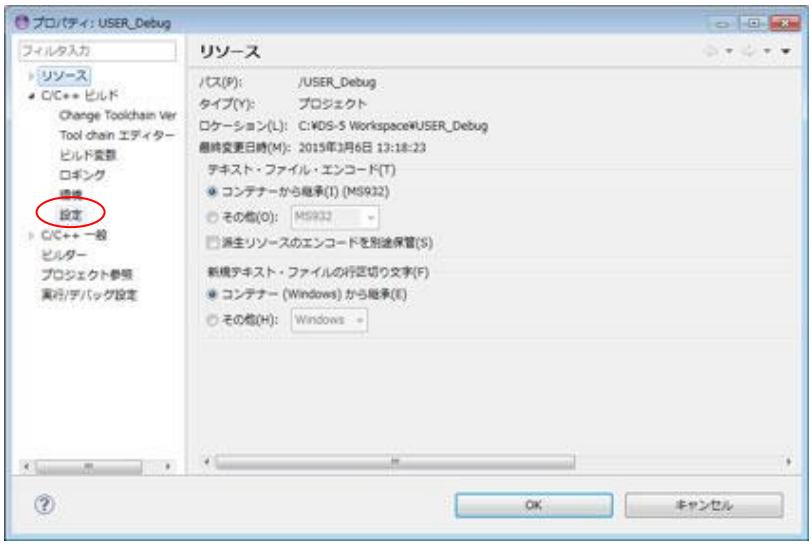
DS-5(ARMC)ツールチェインの設定方法とサンプルプロジェクトに必要な設定を説明します。

１．ツールチェイン設定画面を開きます。

１）【プロジェクト】－【プロパティ】を選択します。

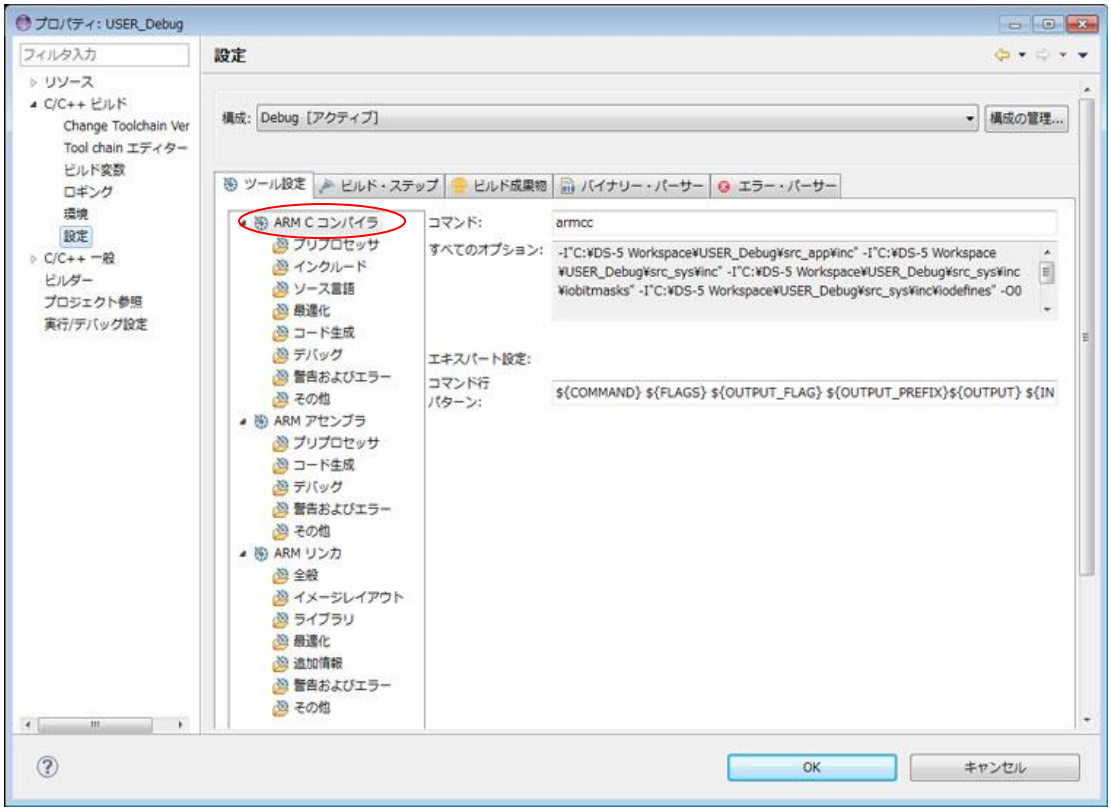


２）「C/C++ビルド」－「設定」を選択します。

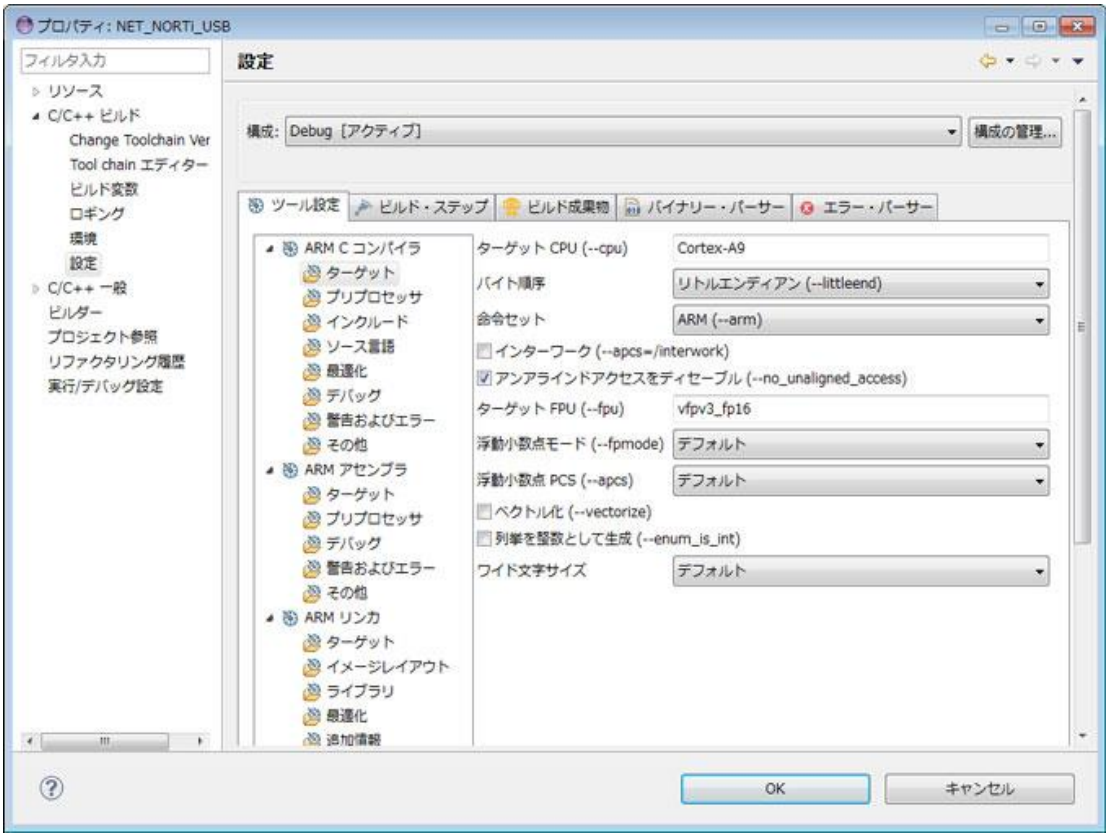


2．ARMC コンパイラの設定

1）ARMC コンパイラのすべてのオプションを表示



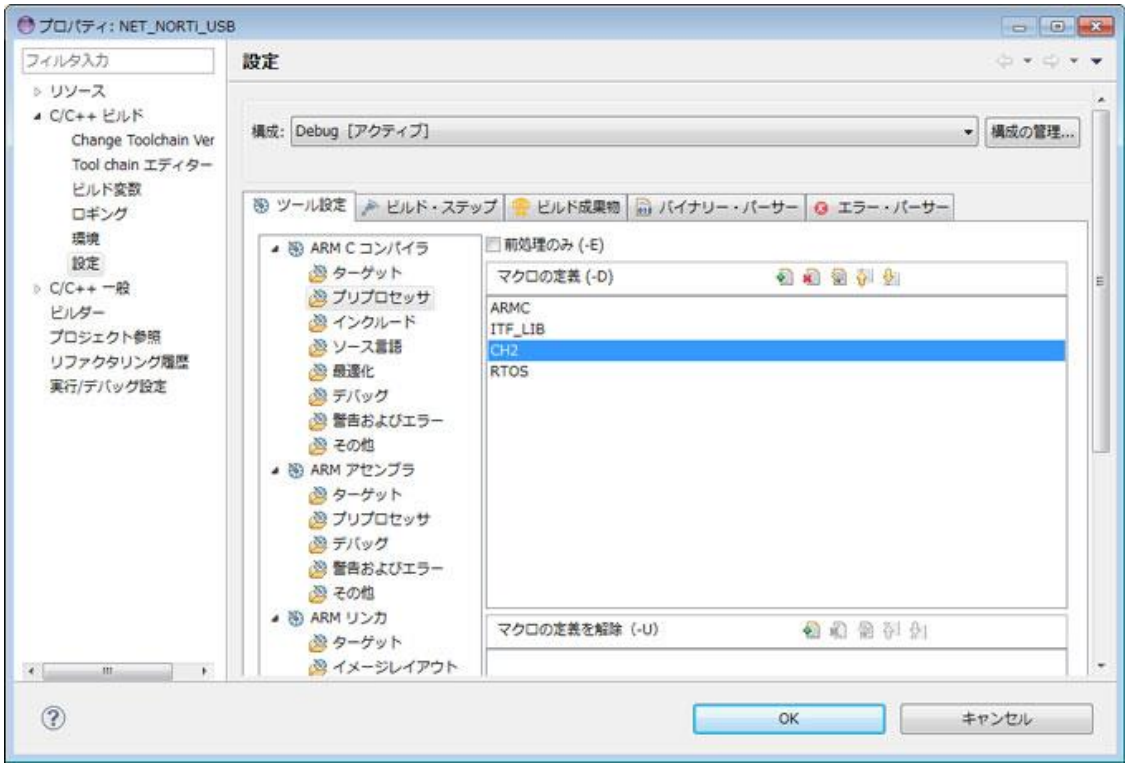
2）ターゲット設定画面



ターゲット CPU(--cpu)	Cortex-A9
バイト順序	リトルエンディアン(--littleend)を選択
命令セット	ARM(--arm)を選択
インターワーク	Thmb 命令のコードを使用する可能性がある場合は、☑して下さい。
アンアライドアクセスをディセーブル	☑
ターゲット FPU(--fpu)	vfpv3_fp16

その他は「デフォルト」設定

3）プリプロセッサ設定画面



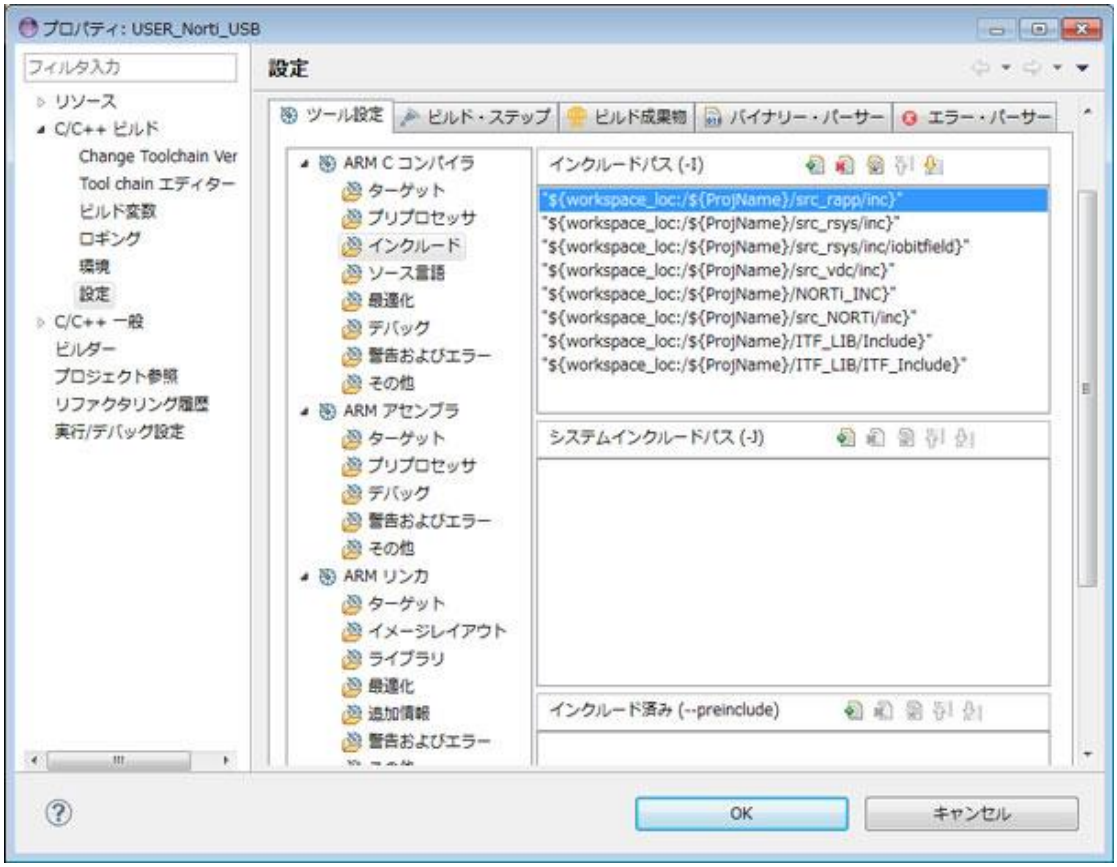
Cソースに#ifdef等のマクロ定義している場合に使用します。	
注＊１ __USED_DEFnano__=x	x＝DEFnanoを使用[1]する・[0]しない。
RTOS	NORTi使用時に定義
CH2	NORTi使用時に定義
ITF_LIB	USB-Function使用時に定義
ARMC	USB-Function使用時に定義

サンプルプロジェクト別に必要なマクロ定義例					
USER_Debug	__USED_DEFnano__=1				
EVrxRZ_Sample	__USED_DEFnano__=1				
EVrxRZ_Sample_USB	__USED_DEFnano__=1			ITF_LIB	ARMC
EVrxRZ_Norti	__USED_DEFnano__=1	RTOS	CH2		
EVrxRZ_Norti_USB	__USED_DEFnano__=1	RTOS	CH2	ITF_LIB	ARMC
EVRZ_Sample	__USED_DEFnano__=1				
EVRZ_Sample_USB	__USED_DEFnano__=1			ITF_LIB	ARMC
EVRZ_Norti	__USED_DEFnano__=1	RTOS	CH2		
EVRZ_Norti_USB	__USED_DEFnano__=1	RTOS	CH2	ITF_LIB	ARMC

注＊１

「__USED_DEFnano__=0」と使用しない側に定義しても内蔵 RAM へのダウンロードとシリアルフラッシュ ROM への書き込み操作は可能です。ただし、再操作する場合はターゲット側のリセット操作が必要になります。

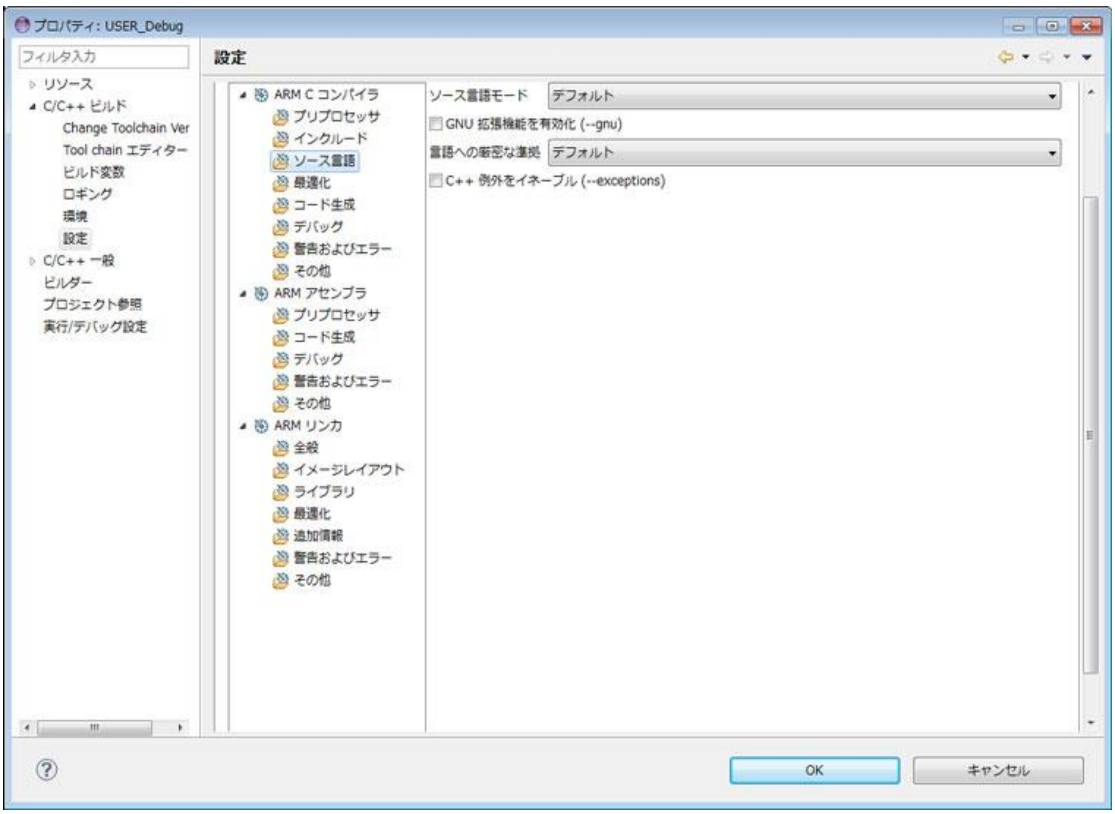
4）インクルード設定画面



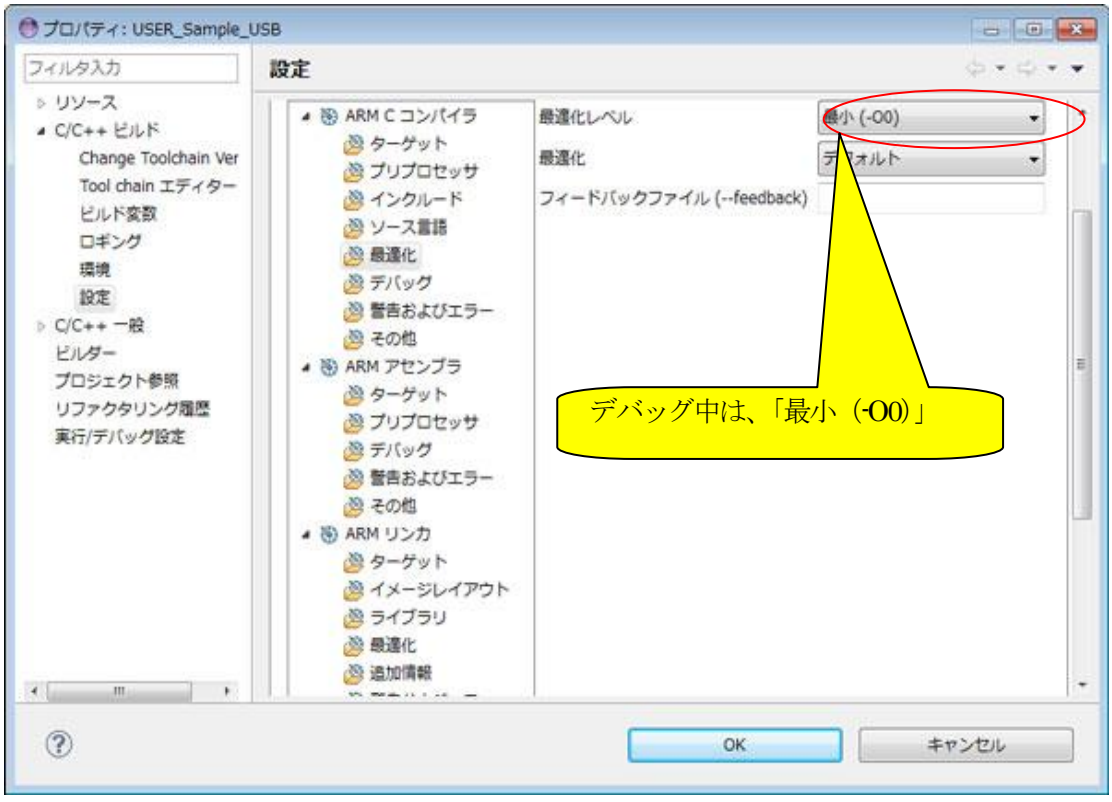
サンプルプロジェクト別に必要なインクルードパスの設定例	
USER_Debug	"${workspace_loc:/${ProjName}/src_app/inc}" "${workspace_loc:/${ProjName}/src_sys/inc}" "${workspace_loc:/${ProjName}/src_sys/inc/iodefines}"
EVrxRZ_Sample	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}"
EVrxRZ_Sample_USB	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/Include}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/ITF_Include}"
EVrxRZ_Norti	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/NORTi/INC}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/NETSMP/inc}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/SMP/inc}"
EVrxRZ_Norti_USB	追加+ "${workspace_loc:/${ProjName}/ITF_LIB/Include}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/ITF_Include}"
EVRZ_Sample	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/src_evb/inc}" 追加+ "${workspace_loc:/${ProjName}/src_vdc/inc}"
EVRZ_Sample_USB	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/src_evb/inc}" 追加+ "${workspace_loc:/${ProjName}/src_vdc/inc}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/Include}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/ITF_Include}"
EVRZ_Norti	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/src_evb/inc}" 追加+ "${workspace_loc:/${ProjName}/src_vdc/inc}"

	追加+ "${workspace_loc:/${ProjName}/NORTi/INC}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/NETSMP/inc}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/SMP/inc}"
EVRZ_Norti_USB	追加+ "${workspace_loc:/${ProjName}/src_eva/inc}" 追加+ "${workspace_loc:/${ProjName}/src_evb/inc}" 追加+ "${workspace_loc:/${ProjName}/src_vdc/inc}" 追加+ "${workspace_loc:/${ProjName}/NORTi/INC}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/NETSMP/inc}" 追加+ "${workspace_loc:/${ProjName}/NORTi_smp/SMP/inc}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/Include}" 追加+ "${workspace_loc:/${ProjName}/ITF_LIB/ITF_Include}"

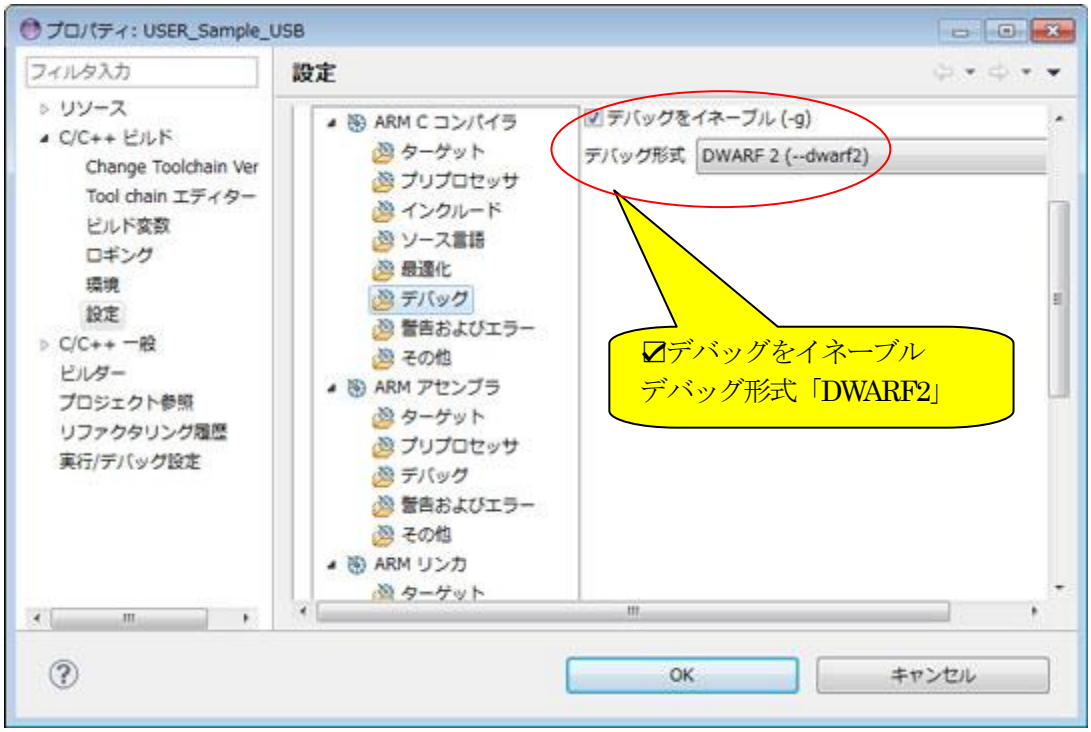
5）ソース言語設定画面（デフォルト設定）



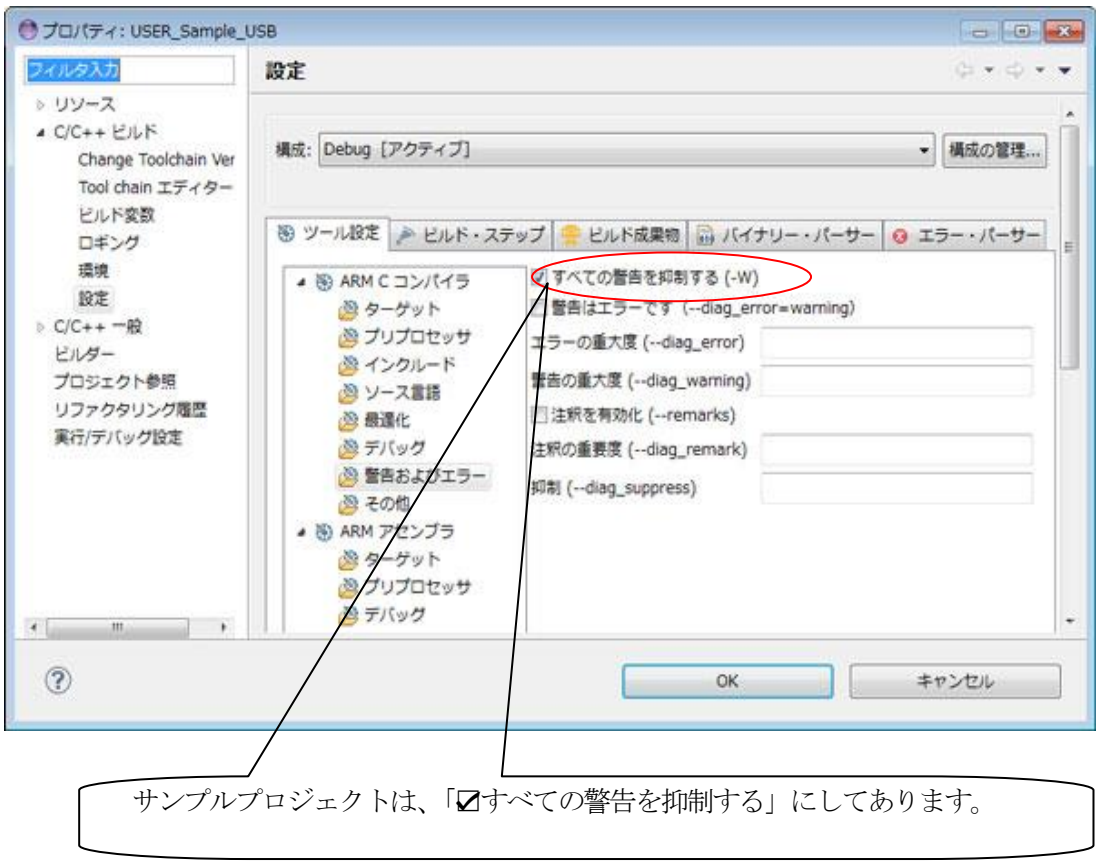
6）最適化設定画面



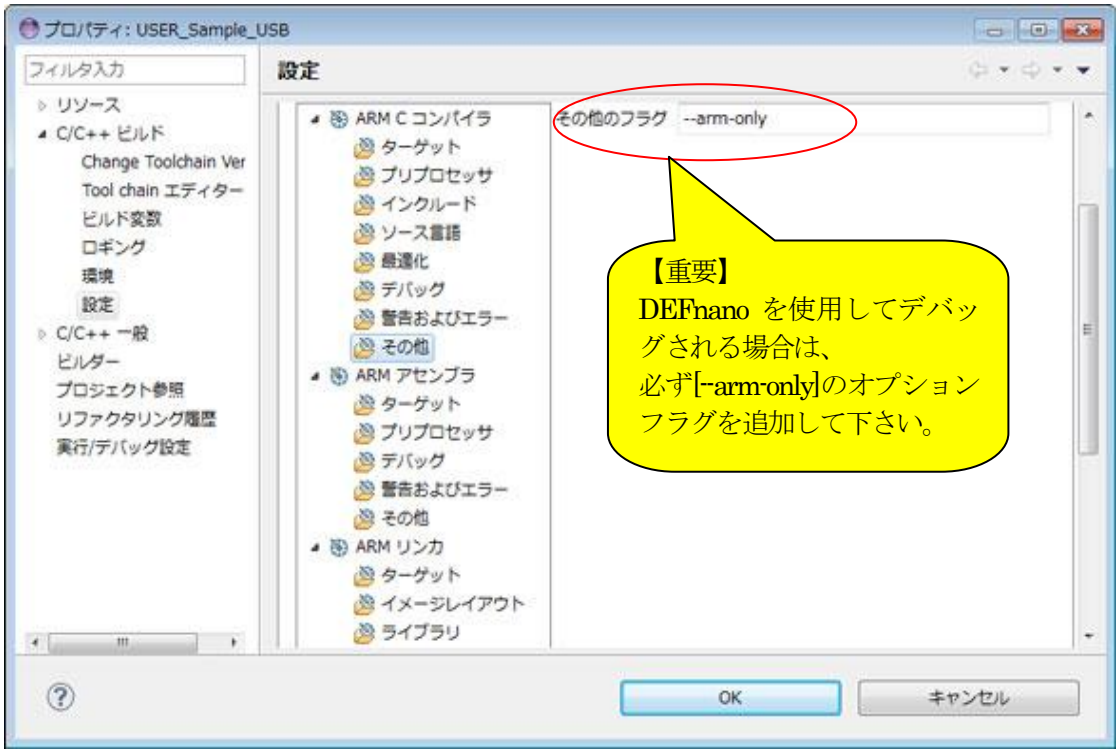
7）デバッグ設定画面



8）警告およびエラー設定画面

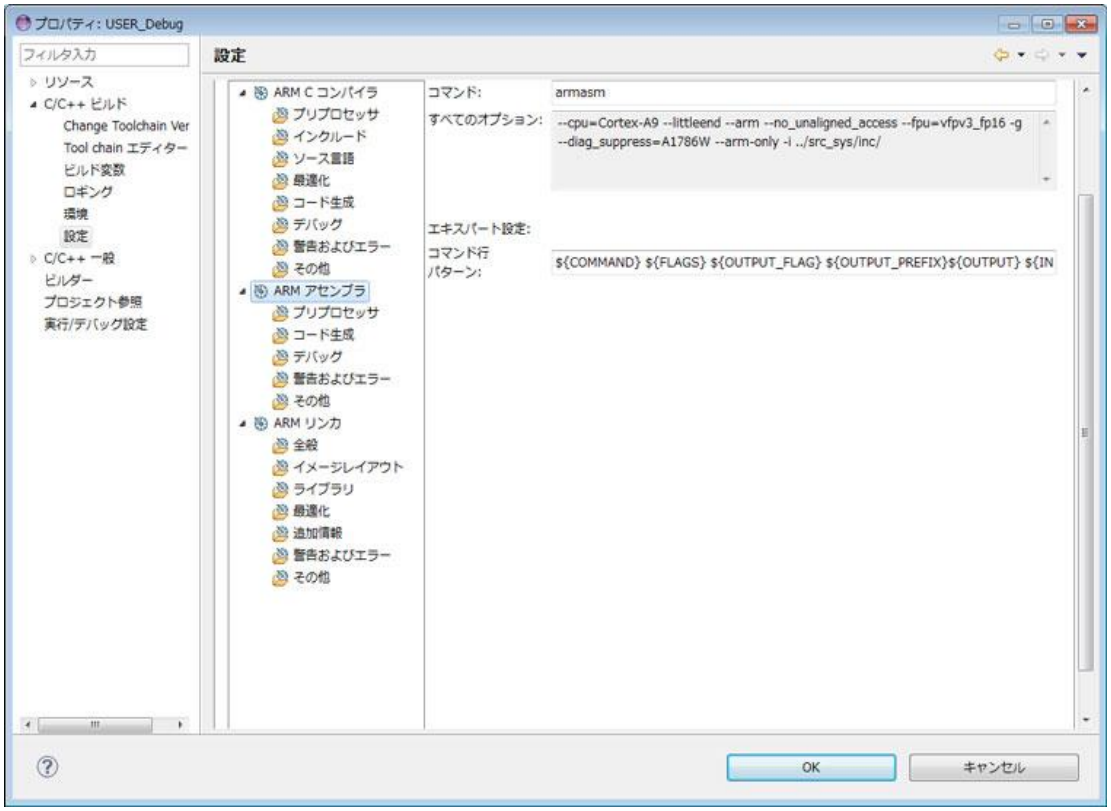


9）その他設定画面

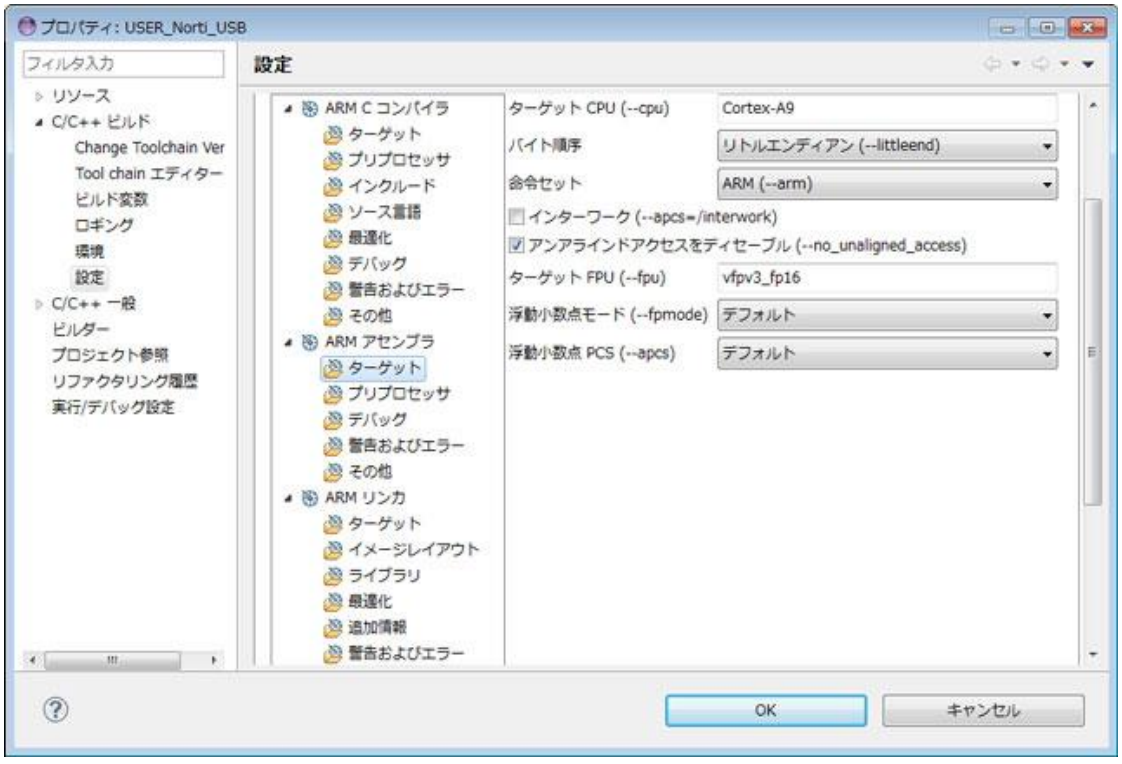

プロパティ: USER_Sample_USB
フィルタ入力
リソース
C/C++ ビルド
Change Toolchain Ver
Tool chain エディター
ビルド変数
ロギング
環境
設定
C/C++ 一般
ビルダー
プロジェクト参照
リファクタリング履歴
実行/デバッグ設定
設定
ARM C コンパイラ
ターゲット
プリプロセッサ
インクルード
ソース言語
最適化
デバッグ
警告およびエラー
その他
ARM アセンブラ
ターゲット
プリプロセッサ
デバッグ
警告およびエラー
その他
ARM リンカ
ターゲット
イメージレイアウト
ライブラリ
その他のフラグ
--arm-only
【重要】
DEFnano を使用してデバッグされる場合は、
必ず[-arm-only]のオプションフラグを追加して下さい。
OK
キャンセル

3．ARM アセンブラの設定

1）ARM アセンブラのすべてのオプションを表示



2）ターゲット設定画面

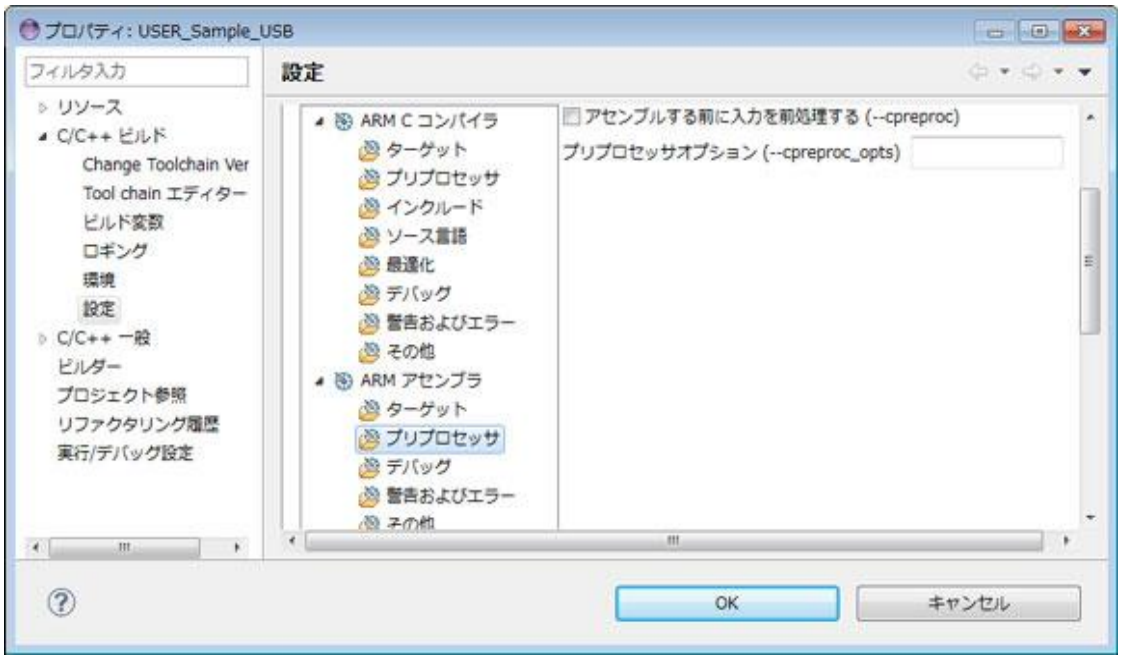


ARMC コンパイラのターゲット設定と同じにする。

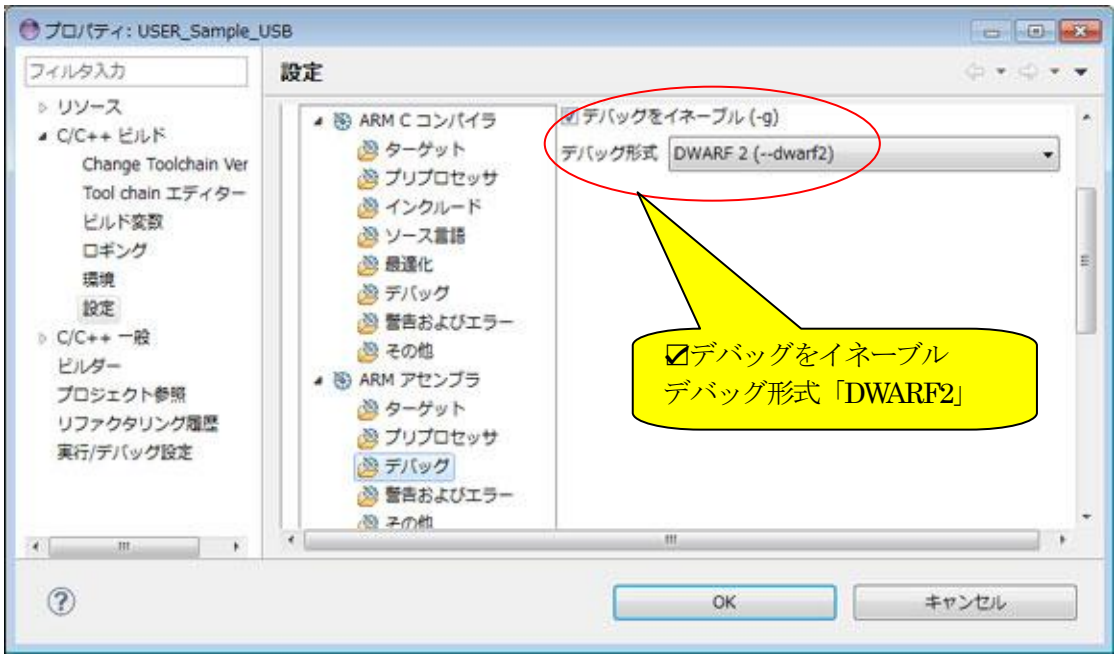
ターゲット CPU(--cpu)	Cortex-A9
バイト順序	リトルエンディアン(--littleend)を選択
命令セット	ARM(--arm)を選択
インターワーク	Thmb 命令のコードを使用する可能性がある場合は、☑して下さい。
アンアライドアクセスをディセーブル	☑
ターゲット FPU(--fpu)	vfpv3_fp16

その他は「デフォルト」設定

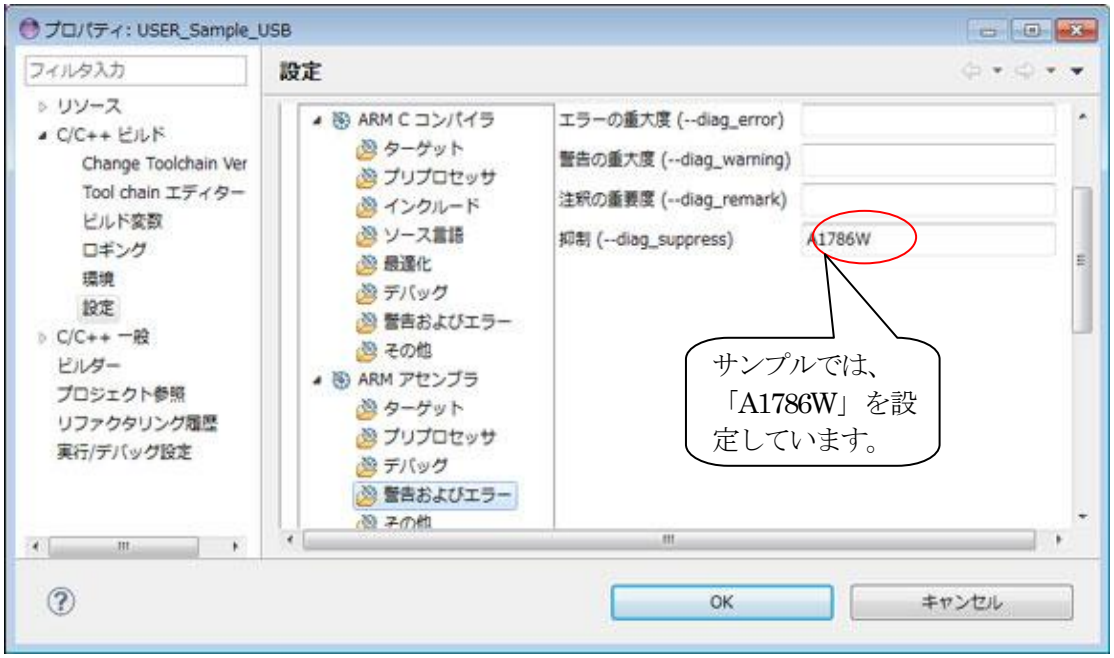
3）プリプロセッサ設定画面（デフォルト設定）



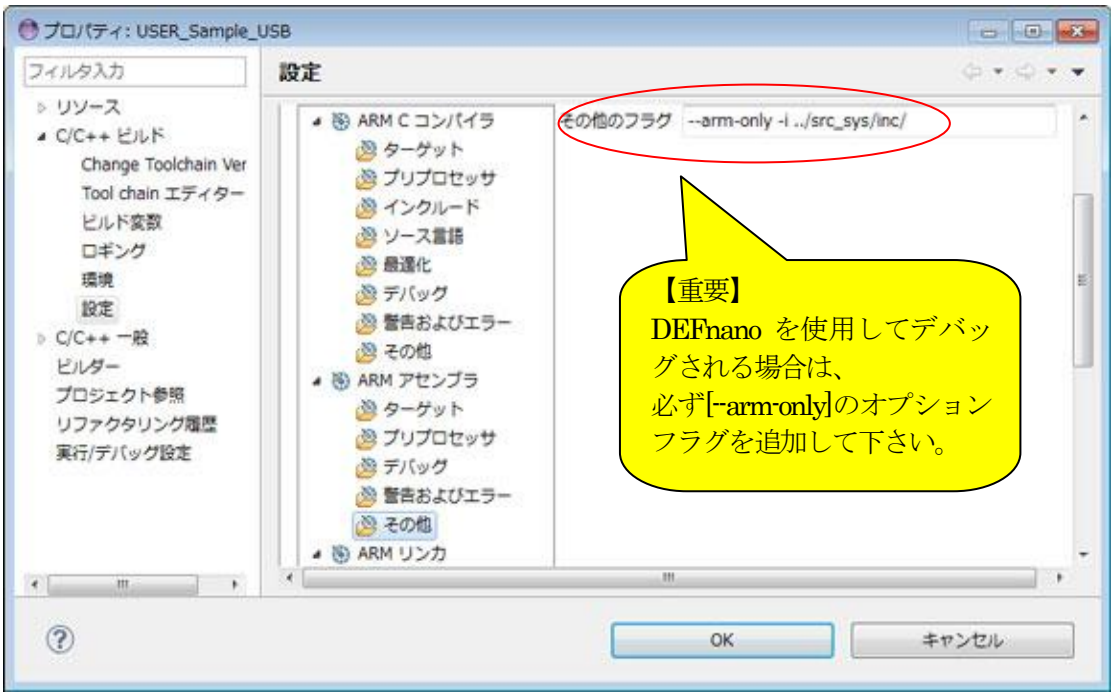
4）デバッグ設定画面



5）警告およびエラー設定画面



6）その他設定画面



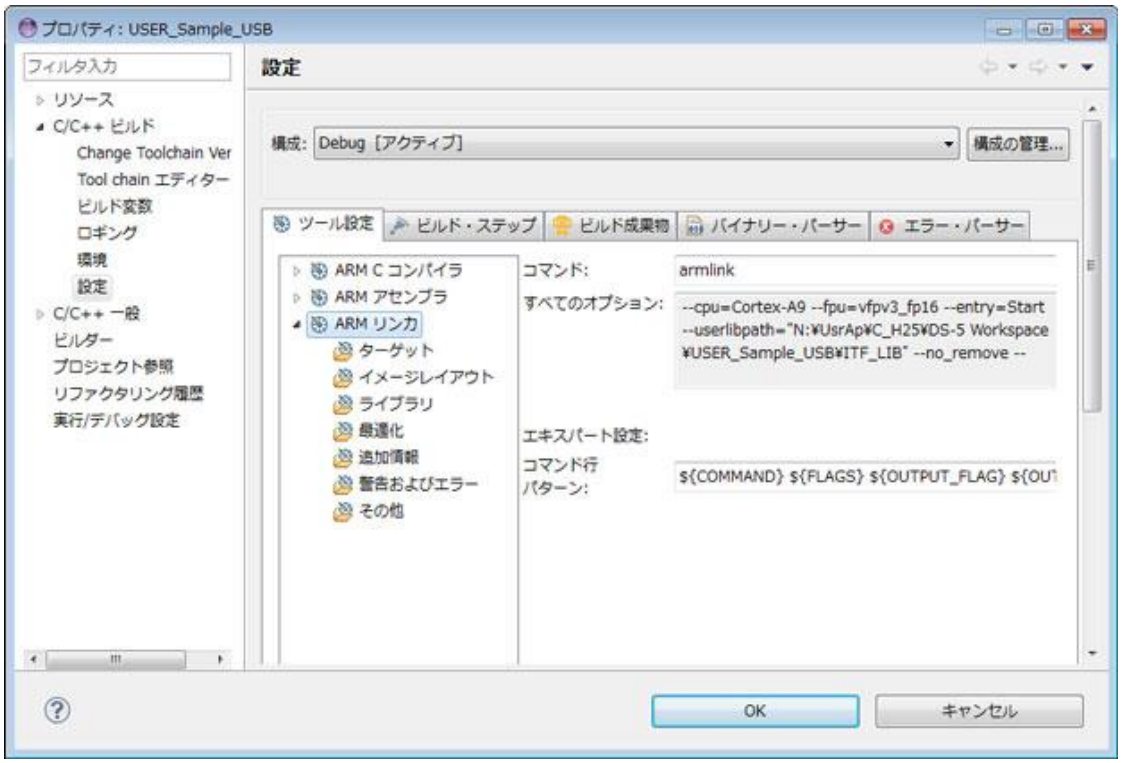
サンプルプロジェクトの追加オプション		
USER_Debug EVrxRZ_Sample EVrxRZ_Sample_USB EVRZ_Sample EVRZ_Sample_USB	--arm-only -i ../src_sys/inc/ 注＊１ --pd="__USED_DEFnano__ EQU x"	ARM_ONLY コード インクルードパスの設定 x＝ DEFnano を 使用[1]する・[0]しない。
EVrxRZ_Norti EVrxRZ_Norti_USB EVRZ_Norti EVRZ_Norti_USB	--arm-only --pd="MMU SETL {TRUE}" --pd="CACHE_ISETL {TRUE}" --pd="CACHE_D SETL {TRUE}" --pd="CACHE_L2 SETL {TRUE}" --pd="CACHE_WT SETL {TRUE}" 注＊１ --pd="__USED_DEFnano__ EQU x"	ARM_ONLY コード MMU と各キャッシュを有効・無効の指定用フラグ x＝ DEFnano を 使用[1]する・[0]しない。

注＊１

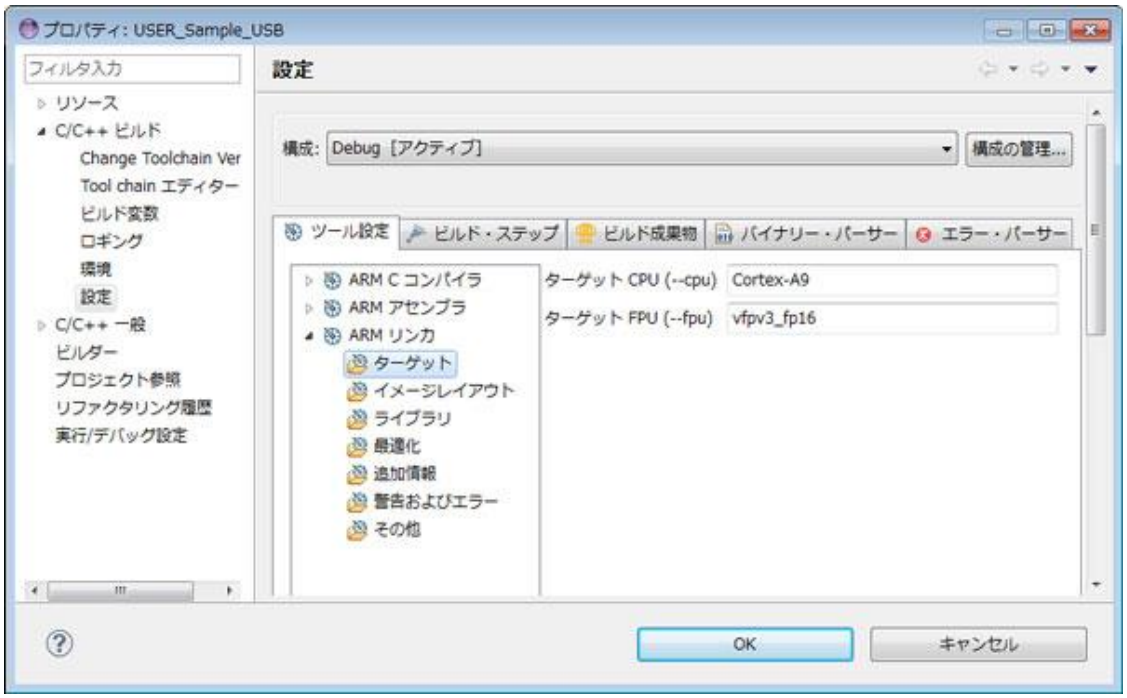
「__USED_DEFnano__ EQU 0」と使用しない側に定義しても内蔵 RAM へのダウンロードとシリアルフラッシュ ROM への書き込み操作は可能です。ただし、再操作する場合はターゲット側のリセット操作が必要になります。

4．ARM リンカの設定

1）ARM リンカのすべてのオプションを表示



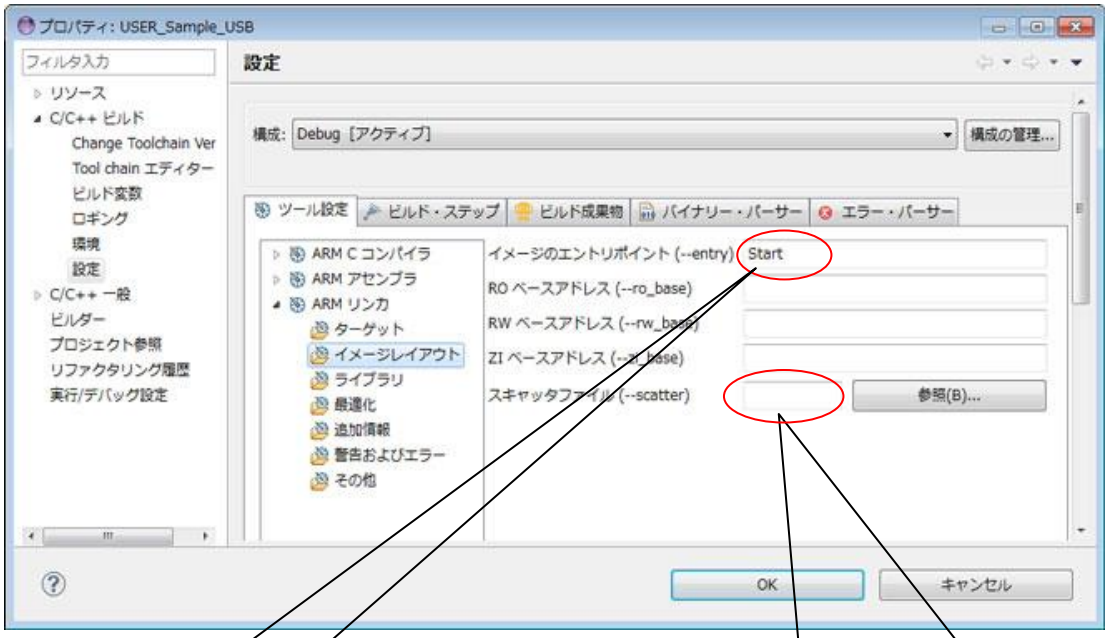
2）ターゲット設定画面



ARMC コンパイラと ARM アセンブラのターゲット設定と同じにする。

ターゲット CPU(--cpu)	Cortex-A9
ターゲット FPU(--fpu)	vfpv3_fp16

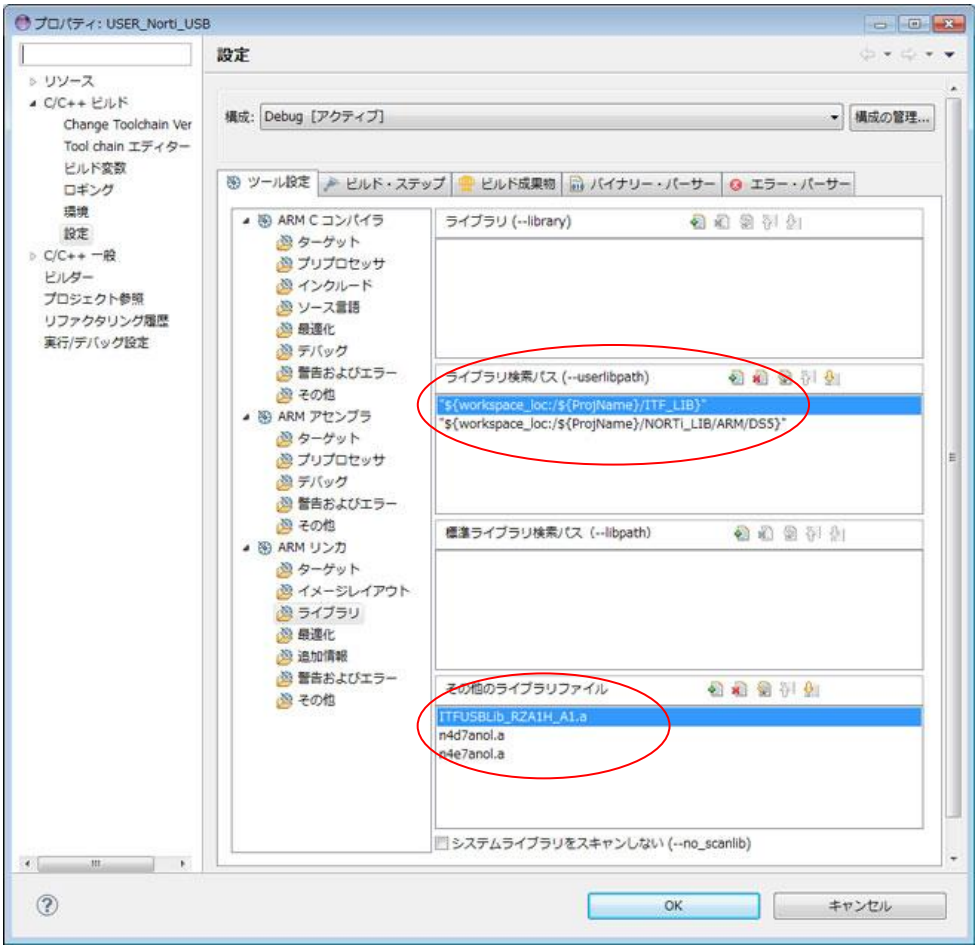
3）イメージレイアウト設定画面



ベクターテーブルの先頭シンボルを設定する。

⚠スキャッタファイルをここで間接指定するとビルドエラーになるため設定しない。(DS-5 Ver5.20.2)
対策として、その他設定で追加する。

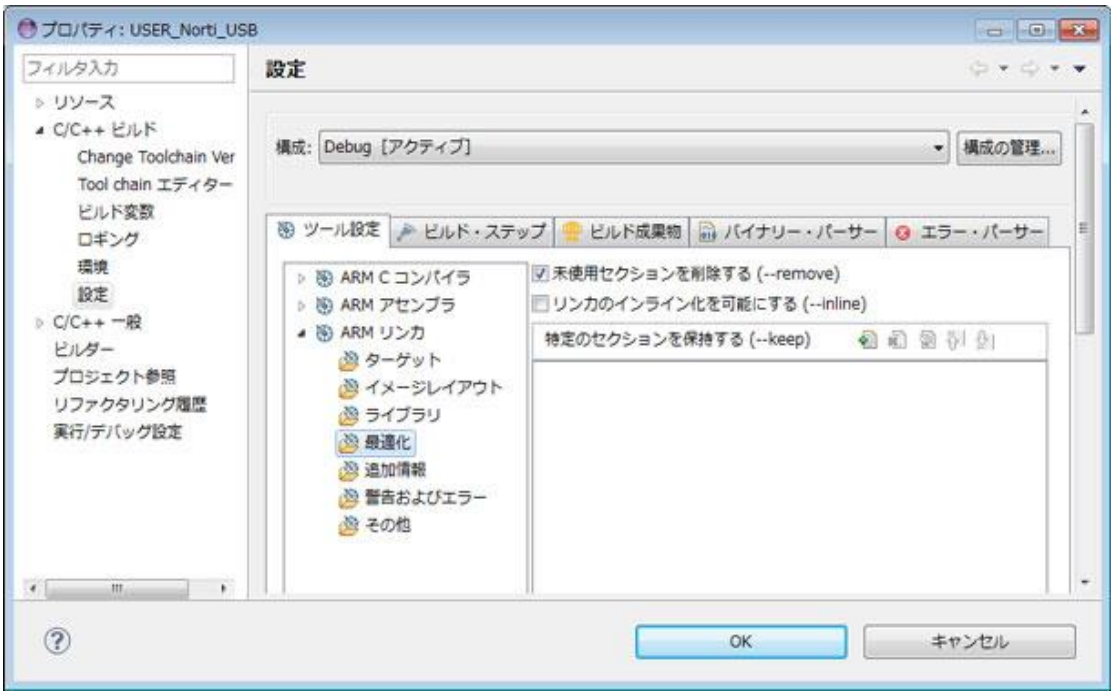
4）ライブラリ設定画面



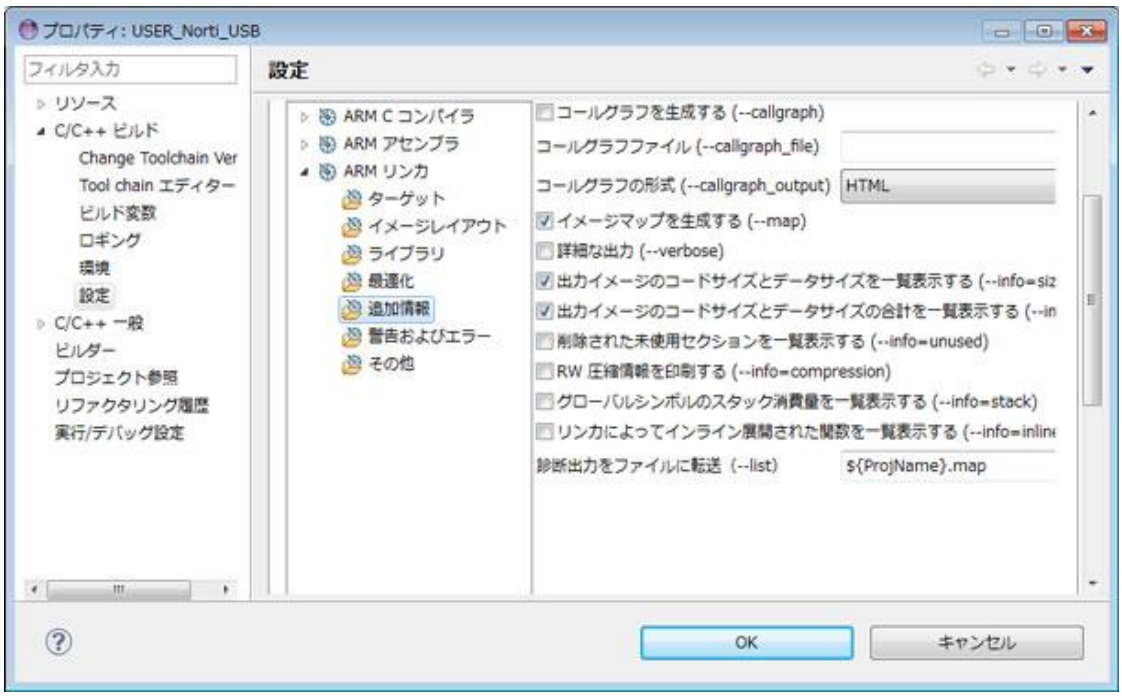
サンプルプロジェクト別のライブラリ検索パス	
USER_Debug EVrxRZ_Sample EVRZ_Sample	なし
EVrxRZ_Sample_USB EVRZ_Sample_USB	"${workspace_loc:/${ProjName}/ITF_LIB}"
EVrxRZ_Norti EVRZ_Nort	"${workspace_loc:/${ProjName}/NORTi_LIB/CA9/DS5}"
EVrxRZ_Norti_USB EVRZ_Norti_USB	"${workspace_loc:/${ProjName}/ITF_LIB}" "${workspace_loc:/${ProjName}/NORTi_LIB/CA9/DS5}"

サンプルプロジェクト別のライブラリファイル	
USER_Debug EVrxRZ_Sample EVRZ_Sample	なし
EVrxRZ_Sample_USB EVRZ_Sample_USB	ITFUSBLib_RZA1H_A1.a（別売り）
EVrxRZ_Norti EVRZ_Norti	n4d7anol.a（別売り） n4e7anol.a（別売り）
EVrxRZ_Norti_USB EVRZ_Norti_USB	ITFUSBLib_RZA1H_A1.a（別売り） n4d7anol.a（別売り） n4e7anol.a（別売り）

5）最適化設定画面（デフォルト設定）

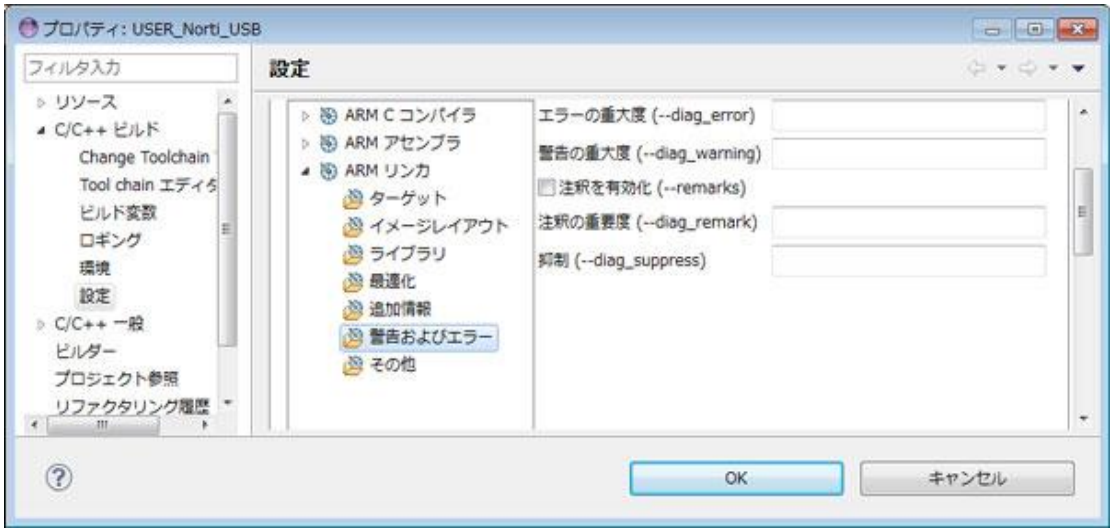


6）追加情報の設定画面

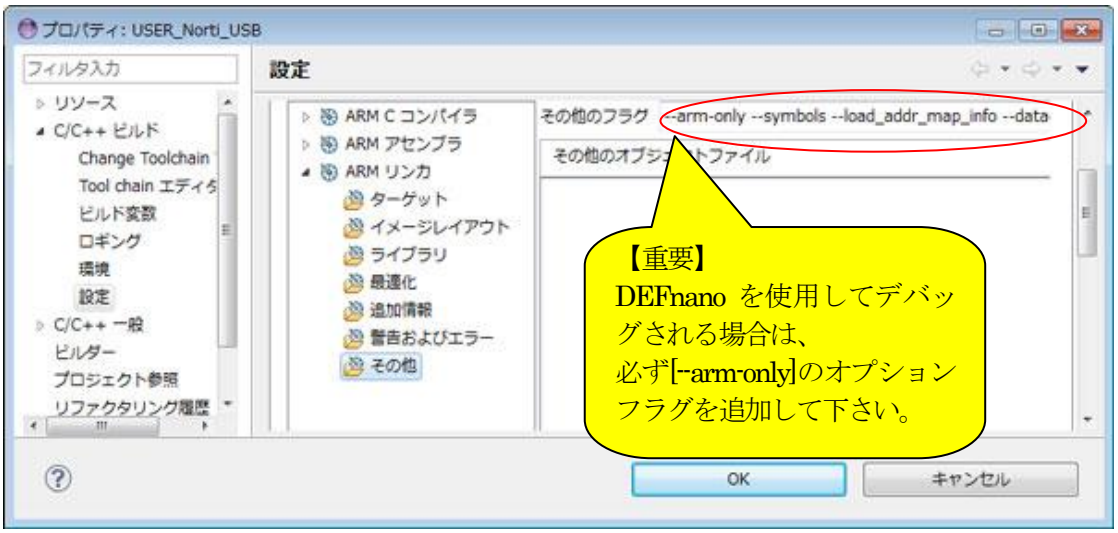


リンカで生成される「Project.map」に追加される情報の選択画面です。必要に応じて選択する。

7）警告およびエラー設定画面（デフォルト設定）

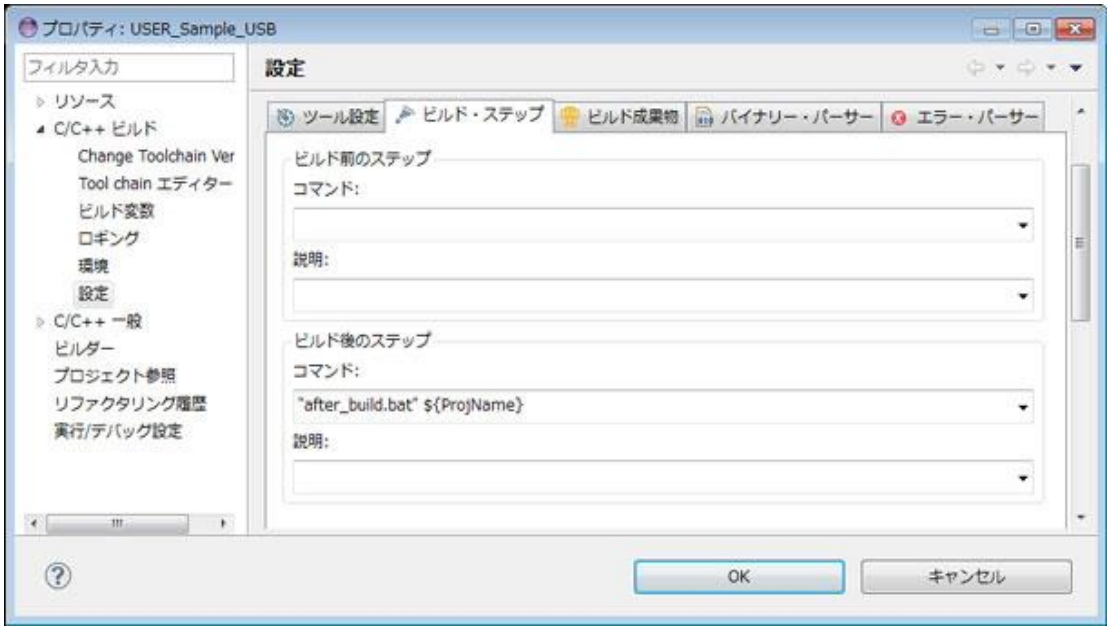


8）その他設定画面



追加オプション	
--arm-only	標準ライブラリを ARM 命令のみで作成
--symbols	各ローカルとグローバルシンボル、およびそのアドレス一覧を作成する。
--load_addr_map_info	実行領域のロードアドレスをマップファイルに追加する。
--datacompressor=off	RW データ圧縮を無効にする。
--library_type=standardlib	リンク時にフルランタイムライブラリを選択します。
--scatter ..¥scatter_file¥scatter.scat	各セクションのアドレス指定にスキャッタファイルを使用する。

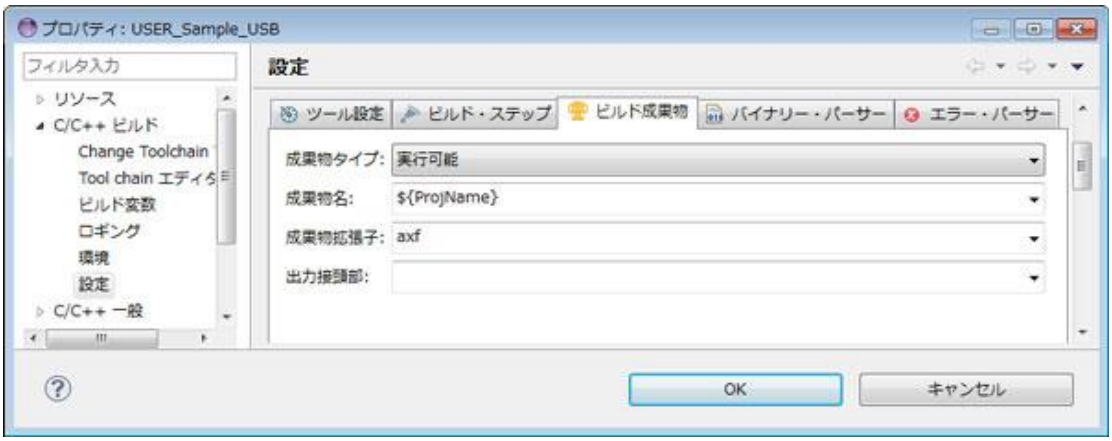
5．ビルド・ステップの設定画面



ビルド実行前と実行後に追加したいコマンドがある場合に設定します。

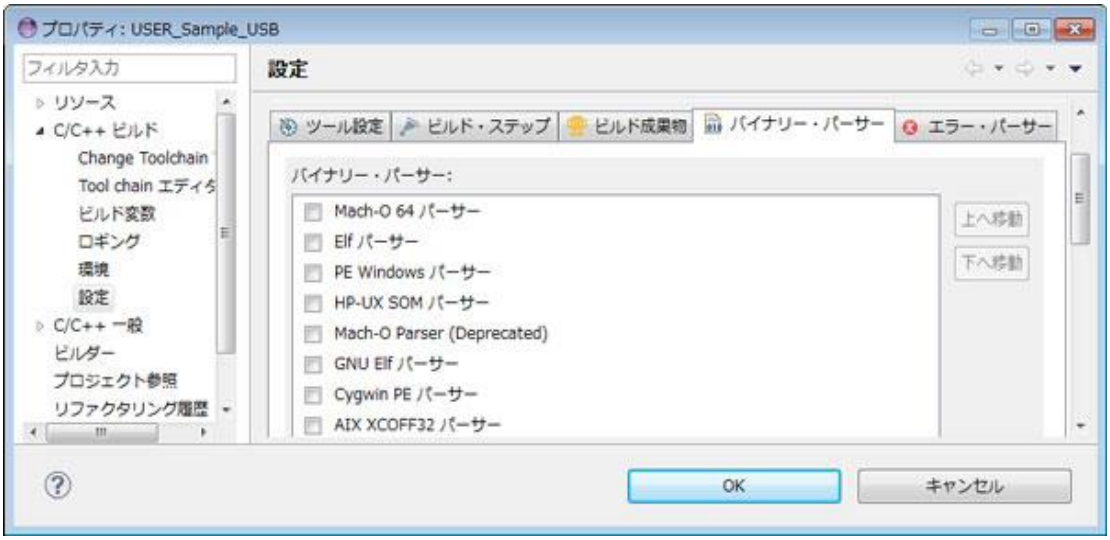
サンプルプロジェクトでは「after_build.bat」にて実行後に設定する。	
fromelf --bin --output=%1.bin %1.axf	bin ファイルの作成
fromelf --m32combined --output=%1.mot %1.axf	mot ファイルの作成

6．ビルド成果物の設定画面（デフォルト設定）



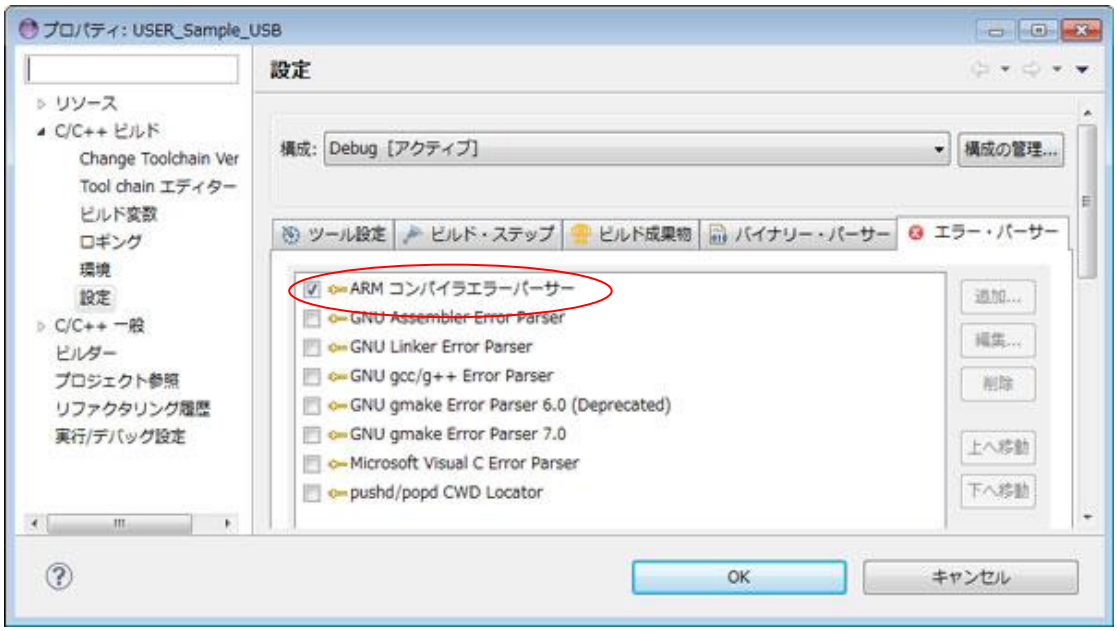
サンプルプロジェクトの設定（DEFnano を使用する場合は、変更不可）	
成果物タイプ	実行可能に選択
成果物名	${ProjName}
成果物拡張子	axf

7．バイナリー・パーサー設定画面（デフォルト設定）



全てノーチェック（使用していません）

8．エラー・パーサー設定画面



「☑ARM コンパイラエラーパーサー」のみを使用する。

9．スキャッタファイルについて

スキャッタファイルとは、セグメントごとにアブソリュートアドレスを定義するためのファイルです。スキャッタファイル独自の予約語がありますので ARM 社が提供しているドキュメント「ARM　コンパイラツールチェーン　リンカの使用：DUI0474GJ_using_the_arm_linker.pdf」の「８：スキャッタファイルの使用」を参照して下さい。

サンプルプロジェクト[scatter.scat]の定義例１

```
#! armcc -E
;Content-Type: text/plain; charset=utf-8
;/*****************************************************************************
;* File Name    : scatter.scat
;* Description : Scatter file
;*****************************************************************************/
;; Definition of On-chip large-capacity RAM (LRAM) area
#define LRAM_START                      (0x20000000)
#define LRAM_SIZE                       (0x00A00000)
#define LRAM_END                        (LRAM_START+LRAM_SIZE)

#define APP_START                       (0x20080000)
#define VECT_SIZE                       (0x100)
#define APP_SIZE                        (0x300000)

#define TTB_START                       ((APP_START+APP_SIZE) AND 0xFFFFC000)
#define TTB_SIZE                        (0x4000)
#define ARM_LIB_STACK_SIZE              (0x8000)
#define STACK_SIZE                      (0x80000)
#define ARM_LIB_HEAP_SIZE               (0x8000)

LOAD_ROM  APP_START         APP_SIZE            ;; CODE/CONST/DATA Area
{
    VECTOR_TABLE            +0          FIXED   ;; CODE/CONST/DATA Area
    { * (VECTOR_TABLE, +FIRST)   }              ;; Vector table

    RESET_INIT_HANDLER      +VECT_SIZE  FIXED
    { * (RESET_INIT_HANDLER)     }

    InRoot                  +0          FIXED
    { * (InRoot$$Sections)       }              ;; All (library) code

    CODE                    +0          FIXED
    { * (+RO-CODE)               }

    CONST                   +0          FIXED
    { * (+RO-DATA)               }

    DATA                    +0          FIXED
    { * (+RW)                    }
```

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;    use as RAM Area       ;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    TTB   TTB_START     EMPTY  TTB_SIZE  ;; Level-1 Table for MMU(AND 0xFFFFC000)
    { }

    ARM_LIB_STACK   +0  EMPTY  ARM_LIB_STACK_SIZE   ;; Application stack
    { }

    STACK_TOP       +0  EMPTY  STACK_SIZE
    { }

    ARM_LIB_HEAP    +0  EMPTY  ARM_LIB_HEAP_SIZE    ;; Application heap
    { }

    BSS             +0
    { * (+ZI)        }

    LAN_BSS          0x60700000                     ;; No Cache Area
    { * (LAN_BSS)    }
    VIDEO_BSS        0x60800000
    { * (VIDEO_BSS)  }
}
```

サンプルプロジェクト[scatter.scat]の定義例の一部

```
    ARM_LIB_STACK STACK_BSS_START     EMPTY  0x8000   ;; Application stack
    { }
    IRQ_STACK               +0        EMPTY  0x8000   ;; IRQ mode stack
    { }
    FIQ_STACK               +0        EMPTY  0x2000   ;; FRQ mode stack
    { }
    SVC_STACK               +0        EMPTY  0x2000   ;; SVC mode stack
    { }
    ABT_STACK               +0        EMPTY  0x2000   ;; ABT mode stack
    { }
    ARM_LIB_HEAP            +0        EMPTY  0x8000   ;; Application heap
    { }
```

EMPTY　0xnnnn は、SIZE を指定していることになります。

スキャッタファイルを編集する場合の注意点

1）#define 行にコメント文をいれないで下さい。
2）漢字入力した場合、このファイルに限り「UTF-8」コードになります。
3）MP-RZA1H 基板を使用する限り、変更する箇所は#define の[VECT_SIZE と TTB_SIZE]以外の xxxx_SIZE のみにして下さい。

１０．ベクターテーブルとローダーの関係について

MP-RZA1H 基板は、シリアルフラッシュ ROM にローダーとアプリケーションプログラムを記憶させ、電源 ON 時にローダーが RZA1H の内蔵 RAM にアプリケーションプログラムをロードして実行させる仕組みになっています。ローダーは内蔵 RAM にロードする時にロード先の先頭アドレスと最終アドレスと実行開始アドレスを知る必要があります。この情報を得るため独自の定義が必要なため下記に説明します。

１）ローダーが必要な情報はベクターテーブルに登録する。「_vector_table_s.s/_vevtor_table_r.s」

```
Start
;=========================================================================
;   Entry point for the Reset handler
;=========================================================================
vector_table
    LDR pc, =Reset_handler          ; Start+0x0000 : リセット
    LDR pc, =Undefined_handler      ; Start+0x0004 : 未定義命令
    LDR pc, =Svc_handler            ; Start+0x0008 : ソフトウェア割り込み
    LDR pc, =Prefetch_handler       ; Start+0x000C : プリフェッチアボート
    LDR pc, =Abort_handler          ; Start+0x0010 : データアボート
    LDR pc, =Reserved_handler       ; Start+0x0014 : Reserved
    LDR pc, =Irq_handler            ; Start+0x0018 : IRQ
    LDR pc, =Fiq_handler            ; Start+0x001C : FIQ(NMI)
;=========================================================================
;   SFROMに登録してあるローダーに渡す情報
;=========================================================================
Info_table
DCD   |Image$$VECTOR_TABLE$$Base|  ; Start+0x0020 :①内蔵RAM転送先の開始アドレス
DCD   |Image$$DATA$$Limit|         ; Start+0x0024 :②内蔵RAM転送先の終了アドレス(+1)
DCD   vector_table                 ; Start+0x0028 :③初期PC値
DCD   0                            ; Start+0x002C :④デバッグモードフラグ
                                   ;   DEFnanoを未使用にして、 USB0を開放する場合は、
                                   ;   0xDEF0DEF0を定義する。
Info_end
;=========================================================================
```

【重要】
①②③④の情報は、ROM 化するためには必要な情報テーブルです。必ず、定義して下さい。

【注意事項】
④で「DEFnano 未使用」コード「0xDEF0DEF0」をセットし、シリアルフラッシュ ROM に登録した場合、二度と DEFnano を使用することが出来なくなります。復帰したい場合は、JTAG デバッガ等でシリアルフラッシュ ROM アドレス「0x2_002C」を未使用コード「0xDEF0DEF0」以外の数値を直接書き込んでください。

以上です。

１１．注意事項

・本文書の著作権は、エーワン（株）が保有します。
・本文書を無断での転載は一切禁止します。
・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
・本文章に関して、ARM 社およびルネサス　エレクトロニクス社への問い合わせは御遠慮願います。
・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任は負わないものとします。
・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありましたら弊社までご連絡くだされば幸いです。
・本文書の内容は、予告なしに変更されることがあります。

１２．商標

・ARM DS-5 は、ARM 社の登録商標、または商品名称です。
・RZ および RZ/A1H は、ルネサス　エレクトロニクス株式会社の登録商標、または商品名です。
・その他の会社名、製品名は、各社の登録商標または商標です。

１３．参考文献

・「RZ/A1H グループ　ユーザーズマニュアル　ハードウェア編」
ルネサス　エレクトロニクス株式会社
・ルネサス　エレクトロニクス株式会社提供のサンプル集
・「armcc ユーザガイド DUI 0472JJ」　ARM 社
・「アセンブラの使用 DUI 0473GJ」　ARM 社
・「リンカの使用 DUI 0474GJ」　ARM 社
・「コンパイラリファレンスガイド　DUI 0328BJ」　ARM 社
・「アセンブラリファレンス DUI 0489GJ」　ARM 社
・「armkink リファレンスガイド DUI 0804AJ」　ARM 社
・その他

〒486-0852
愛知県春日井市下市場町 6-9-20
エーワン株式会社
http://www.robin-w.com