Rev 1.10.00

EWARM(ICC版)ツールチェインの設定と必要事項の説明

(ルネサス RZ/T1用)

EWARM(ICC版)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。

下記 5 通りのサンプルプログラムがありますが、toolchain の設定例は「RZT1_Sample_BARE」をもとに説明を 進めます。なお、他のサンプル詳細は、サンプル用アプリケーションノートを参照して下さい。

サンプル名	用途	ワークスペース名	Ц	プロジェクト名
			ア	
RZT1_Sample_BARE	ベアメタル版サン	Sample_BARE.eww	M3	Sample_BARE_M3
	プル		R4F	Sample_BARE_R4F
RZT1_Sample_HWRTOS	M3側で	Sample_HWRTOS.eww	M3	Sample_HWRTOS_M3
	HWRIOS を使用 したサンプル		R4F	Sample_HWRTOS_R4F
RZT1_Sample_ECAT	EtherCAT 制御サ	Sample_ECAT.eww	M3	Sample_ECAT_M3.mot を
	シリル			使用(ソース非公開)
			R4F	Sample_ECAT_R4F
RZT1_Demo_BARE	ベアメタル版デモ	Demo_BARE.eww	M3	Sample_ECAT_M3.mot を
	ソフト			使用(<u>ソース非公開</u>)
			R4F	Demo_BARE_R4F
RZT1_Demo_NORTi	RTOS-NORTi 版デ	Demo_NORTi.eww	M3	Sample_ECAT_M3.mot を
	モソフト			使用(<u>ソース非公開</u>)
			R4F	Demo_NORTi_R4F

注記

コア【M3】側 EtherCAT®制御用サンプルのソース公開を希望される場合は、JSL Technology 社とのライセンス契約が必要です。

1. IAR Embedded Workbench IDEを起動する。



1-2. ワークスペースファイルの一例として「Sample_BARE.eww」を選択する。

※ ワークスペースを開く		X
Image: Constraint of the state of the stat	RZT1_Sample_B	AREの検索 🔎
整理 ▼ 新しいフォルダー	: :==	• 🔟 🔞
▲ 🔐 RZT1_Sample_BARE 🔺 名前	更新日時	種類
M3 M3 M3	2017/05/22 11:51	ファイル フォル
R4F	2017/05/22 11:51	ファイル フォル
settings	2017/05/22 11:51	ファイル フォル
src Sample_BARE.eww	2017/05/22 11:50	IAR IDE Works
4 🎍 R4F		
>] Debug		
🔒 settings		
> 🚺 src 🗸 🗸 👘		Þ
Sample_BARE.eww 状況: 32 共有 サイズ: 249 パイト IAR IDE Workspace 更新日時: 2017/05/22 11:50 作成日時: 2017/05/22 11:50)	
ファイル名(N): Sample_BARE.eww ・	ワークスペースファ	′イル(*.ew ▼
	開<(0)	キャンセル



2. 各ツールの設定内容を確認する。

2-1. コア【R4F】側の確認

* main_s.c f0 * × * 1 //***********************************
Image: 1 1
BARE R4 Hブション(0) Sommon
fe BARE R4 オブション(0) 公里 ①「Sample_BARE_R4F」選択 sommon conitor sample put ンパイル(C) ②右クリック Ik(800MHz) すべてを再ビルド(B) クリーン(L) ③「オプション」選択 Deb Ik(800MHz) こSTAT静的解析(C) ・ 注意事項 CMT0ユニットは使用に注意が必要!! r_opg.o/r_ecm.cが使用している。処理後は利用可 道加(A) ・ if ************************************
common tonitor sample put メイク(M) ② ① 「Sample_BARE_R4F] 選択 1k(600MHz) コンパイル(C) sample put コンパイル(C) ③ ③ 「オプション」選択 1k(600MHz) ウリーン(L) Deb ③ 「オプション」選択 1k(600MHz) C-STAT静的解析(C) ・ 注意事項 CMT0ユニットは使用に注意が必要!! r_orgs.c/r_ecm.cが使用している。処理後は利用可 道加(A) ・ ・ 潮除(V) 名前の変更 ・ ノバージョン管理システム(Y) ・ ファイルの場所を開く ファイルのプロパティ(P) ・ マロー、パージャート宣言 ・
ionitor コンパイル(C) (2石クリック (k(8000MHz)) iample マベてを再ビルド(B) () () () yut () マイてを再ビルド(B) () () () C-STAT静的解析(C) ・ 注意事項 CMT0ユニットは使用に注意が必要!! () ビルドを停止(S) ・ 注意事項 CMT0ユニットは使用に注意が必要!! () iam(A) ・ ・ ・ NB(V) 名前の変更 ・ ・ ファイルの場所を開く ファイルのプロパティ(P) ・ ・ マクニッイブレがた(C) ・ ・ ・ マクニッインの場所を開く ・ ・ ・ マクニッイブレがた(C) ・ ・ ・
sample コンパパパワ monodian put すべてを再ビルド(B) クリーン(L) C-STAT静的解析(C) ・ ビルドを停止(S) ・ 追加(A) ・ 削除(V) 名前の変更 パージョン管理システム(Y) ・ ファイルの場所を開く ファイルのプロパティ(P) ファイルのプロパティ(P) マクニックトロー
Juit クリーン(L) Deb C-STAT静的解析(C) ・ 注意事項 CMT0ユニットは使用に注意が必要!! ビルドを停止(S) 注意事項 CMT0ユニットは使用に注意が必要!! 追加(A) ・ 道加(A) ・ 削除(V) 名前の変更 ・ バージョン管理システム(Y) ・ ファイルのプロパティ(P) ・ マクニッグに約支(C) ・
C-STAT静的解析(C) 注意事項 CMT0ユニットは使用に注意が必要!! r_copg.c/r_ecm.cが使用している。処理後は利用可 逆ルドを停止(S) ************************************
C-STAT静的解析(C) ・ E念事項 CMT0ユニットは使用に注意が必要!! ビルドを停止(S) 追加(A) ・ 適加(A) ・ 削除(V) 名前の変更 ノ(ージョン管理システム(Y) ・ ファイルの場所を開く インポート宣言 ファイルのプロパティ(P) ・
ビルドを停止(S) Image: Control and Contreland Control and Control and Control and Co
追加(A) ・ 追加(A) ・ 削除(V) 名前の変更 de "main_s.h" ノ(ージョン管理システム(Y) ・ ファイルの場所を開く ・ ファイルのプロパティ(P) ・ マカニュノブレジャン(C) ・
削除(V) de "Umonitor.h" 名前の変更 define パージョン管理システム(Y) ファイルの場所を開く ファイルのプロパティ(P) インポート宣言 マクニップに称す(C)
名前の変更 def ine パージョン管理システム(Y) , ファイルの場所を開く , ファイルのプロパティ(P) インポート宣言
パージョン管理システム(Y) ・ ファイルの場所を開く ・ ファイルのプロパティ(P) ・
バーション管理システム(Y) → ファイルの場所を開く ファイルのプロパティ(P) マクニュノブーがティ(C)
ファイルの場所を開く ファイルのプロパティ(P) マクニュノブーがティ(C)
ファイルのプロパティ(P)
マクニ ノブージャナクト
アンノイノに設定(に)
29 //
ple_BARE_M3 Sample_BARE_R4F
nple_BARE_M3 Sample_BARE_R4F
nple_BARE_M3 Sample_BARE_R4F (100 // 100 /
28 //===



2-1-1. 一般オプション

1) ターゲット

カテゴリ:	ターゲット 出力 ライブラリ設定 ライブラリオブション MISRA-C:2004 MISRA-C:1998 フロセッサ選択 コア(0) Cortex-R4 ・ ・ ・ ・ デバイス(D) Renesas R7S910018_R4F ・ ・ ・ デバイス(D) Renesas R7S910018_R4F ・ ・ ・ ・ デバイクンモー ・ ド戸U(F) VFPv8 ・ ・ レジスタ(R) 16 ・ ・	

2) 出力

カテゴリ: <u> 一般オプション</u> 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド	ターゲット 出力 ライブラリ設定 ライブラリオブション MISRA-C:2004 MIS 出力ファイル(O) ④ 実行可能ファイル(E)	RA-C:1998
ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP	● ライブラバU) 出力ディレクトリ Debug¥Exe 実行可能ファイル/ライブラリ(A) Debug¥Exe フラビックティレ(B): Debug¥Obj	
GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris	Debug¥Obj IJストファイル(S): Debug¥List	

3) ライブラリ設定

	_			
カテゴリ: 	E	ターゲット 出力 マイブラリ ライブラリ(し) ノーマル 設定ファイル(C): (\$TOOLKIT_DIR\$¥INC¥c¥D ライブラリのスレッドサポーI 「低レベルインタフェースのライ () なし(N)	設定 うイブラリオフション MI	SRA-C:2004 MISRA-C:1998 常の設定を使用します。ロ ール、ファイル記述子サポー ジイト文字なし、strtodでの
I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	•	 ● セミホスティング(S) ● IARブレークポイント(I) 	 ● セミホスティング経由 ● SWO経由 	ОК <i>₹ту</i> ./2µ

4) ライブラリオプション

カテゴリ:	ターゲット 出力 ライブラリ設定 ライブラリオブション MISRA-C:2004 MISRA-C:199 Printfフォーマッタ 自動 フォーマッタの自動選択。 フォーマッタの自動選択。 フォーマッタの自動選択。 ジォーマッタの自動選択。 アオーマックの自動選択。 ブォーマックの自動選択。 アオーマックの自動選択。
-------	---

5) MISRA-C:2004 (デフォルト)

カテゴリ: <u>一般オプション</u> 静静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デパッガ シミュレータ Angel CMSIS DAP GDBサーパ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris	ターゲット 出力 ライブラリ設定 ライブラリオプション MISRA-C:2004 MISRA-C:1998 MISRA-Cを有効にする(E) MISRA-C:2004 MISRA-C:2004 MISRA-C:2004規則をPクライブに設定 なし(N) 必須(R) すべて(A) 「1.1: [required] All code shall conform to ISO 9899:1990 'Programming langu 「1.2: [required] No reliance shall be placed on undefined or unspecified beh 「1.3: [required] No reliance shall be placed on undefined or unspecified beh 「1.4: [required] Multiple compilers and/or languages shall only be used if th 「1.5: [advisory] Floating-point implementations should comply with defined ② 1.1: [required] Assembly language shall be encapsulated and isolated. ③ 2.2: [required] Assembly language shall be not should comply with defined ③ 2.1: [required] Assembly language shall be not should comply with defined ③ 2.1: [required] Assembly language shall be not should comply with defined ④ 2.2: [required] Assembly language shall be not should comply with defined ④ 2.2: [required] Assembly language shall be not should comply with defined ④ 2.2: [required] Assembly language shall be not should comply with defined ④ 1.5: [advisory] Floating-point implementations should comply with defined ④ 2.1: [required] Assembly language shall be not should comply with defined ④ 2.1: [required] Assembly language shall be not should comply with defined ④ 2.1: [required] Assembly language shall be not should comply with defined ④ 2.1: [required] Assembly language shall be not should comply with defined ④ 1.5: [advisory] Floating-point implementations should comply with defined ④ 1.5: [advisory] [
Macraigor	

6) MISRA-C:1998 (デフォルト)

Angel CMSIS DAR	 I. [required] All code shall conform to ISO 9899 standard C, with no exten 2. [advisory] Code written in languages other than C should only be used 3. [advisory] Assembly language functions that are called from C should t 4. [advisory] Provisions should be made for appropriate run-time checking
GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	 5. [required] Only those characters and escape sequences which are defined. 6. [required] Values of character types shall be restricted to a defined and the used 7. [required] Trieranhs shall not be used



2-1-2. 静的解析

1) C-STAT 静的解析(デフォルト)

カテゴリ:	Г		
 一般オプション			
静的解析			
ランタイム解析 C/C++コンパイラ	<	C-STAT静的解析 追加オプション	
アセンブラ		C-STATチェックを避択(S)	
出力コンバータ		0 81H191992/28/((0/	
カスタムビルド		設定をインポート(1)…	
ビルドアクション	=	設定をエカ7 ポート(F)	
リンカ ゴバッザ		EXECTORY ICT	
シミュレータ		Enable parallel analysis:	2 processes
Angel		🗹 Enable module timeout:	600 seconds
GDBサーバ		Enable false-positives analysis	
IAR ROMモニタ		☑ Limit messages per check and file:	100 messages
I-jet/JTAGjet		この制具バージョンで使用可能から-STATの3	5/+2 ¹ /2 1/1×10+++ /.
J-LINK/J-Trace			パビノヘルのりみと708
11 StellarIS			

2)追加オプション(デフォルト)

カテゴリ:					
一般オブション +245-2044					
高部の3時#40 ランタイム解析 C/C++コンパイラ		C-STAT書的角架析	(追加オプション		
アセンブラ		- コマンドラインオ	ブションの使用(U)		
出力コンバータ		コマンドラインオ:	プション(0):(1行に1コマンド)	
カスタムビルド					
ビルトアクソヨン	E				
ッシン デバッガ					
シミュレータ					
Angel					
CMSIS DAP					
GDBサーバ					
IAR ROMELS					
I-jet/JTAGjet					$\overline{\nabla}$
J-Link/J-Trace					
TI Stellaris					

2-1-3. ランタイム解析

1) C-RUN ランタイム解析 (デフォルト)





2-1-4. C/C++コンパイラ

1) 言語1

i.

カテゴリ: 一般オプション 静的解析 ランタイム解析		■ 複数ファイルのコンパイル 未使用パブリックを破棄 言語 1 言語 2 コード 最適化 2	工場出荷時設定 出力 リスト プリプロセッサ 診断 MISF ・
マセンブラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I.jet/JTAGjet J-Link/J-Trace TI Stellaris	Ε	 書語(1) ● C ● C++ ● 自動(拡張子ベース)(A) CO派生言語(C) ● C89 ● C99 ■ VLAの許可(V) ■ C++ インライン動作 ■ プロトタイプの強制(R) 	 言語の適合(N) ● 標準(IAR拡張あり) ● 標準 ● 厳密 C++の派生言語(+) ● Embedded C++ ● 拡張Embedded C++ ● C++ ● C++ ■ 例外あり ♥ 同外あり ♥ 副的オブジェクトを破棄

2) 言語2

 一般オプション	▲ ● 複数ファイ 二 未供 言語 1 言語 1 言語 1 言語 1 言語 1 言語 1 『 で にね。" (○ 符号 ③ 符号 ③ 第 一 次 1 一 、 (一 、 (明) の 一 不 (明) の 一 、 (明) の 一 、 (明) の 一 、 (の) の 一 、 (の) の 一 、 (の) の) の 、 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 、 、 の 、 、 の の 、 の 、 の 、 の 、 の 、 の 、 、 、 、 、 、 、 、 、 、 、 、 、	ルのコンパイル 和/ブリックを破棄 語2)コード 最適化 出 の型(P) ・付き な点数動作(F) な点数動作(F) にな適合 (サイズ縮小/高速化) ドイト文字サポートを有効にす	カ リスト ブ	 リプロセッサ 診断	(何時話文元 MISF 4))
----------------------------	---	---	---------	-----------------	--------------------

3) コード

カテゴリ:			工場出荷時設定
ー般オブション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ		複数ファイルのコンパイル □ 未使用パブリックを破菜 語 1 言語 2 (ユード) 最適化 出力 リスト	プリプロセッサ 診断 MISF ・
、ビンフンバータ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ	=	□ インタワークコードを生成(1) プロセッサモード ④ Arm(A) ● Thumb(T)	
Angel CMSIS DAP GDBサーバ IAR ROMモニタ		位置独立コード/データ コードおよびリードオンリのデータ (ropi)(C) 回リード/ライトデータ (rwpi)(R) 回動的なリード/ライト初期化なし(N)	
I-jet/JIAGjet J-Link/J-Trace TI Stellaris Macraigor		□]~FXモリハのナ~\$Read+±b%(N)	

4) 最適化



5) 出力

→ 取オブション = 一般オブション 自動的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デパッガ シミュレータ Angel CMSIS DAP GDBサーパ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	 【複数ファイルのコンパイル 未使用パブリックを破棄 言語 1 言語 2 コード 最適化 デバッグ情報の生成(D) codeセクジョン名(C): text .text 	<u>工場出荷時設定</u> <u>出力</u> リスト プリプロセッサ 診断 MISF ・
--	--	--

6) リスト

カテゴリ: 一般オプション 素語の句音245	▲ 【 複数ファイルのコンパイル 】 二場出荷	時設定
ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	■ I 言語 2 コード 最適化 出力 リスト フリプロセッサ 診断 リストファイルの出力(0) ○ アセンブラニーモニック(A) ○ ジザ(D) ■ アセンブラファイルの出力(U) ○ ソースのインクルード(N) ■ 呼出しフレーム情報のインクルード(N)	
Macraigor PE micro	- OK *	ャンセル

7) プリプロセッサ



<mark>注*1</mark>

「_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。

8)診断 (デフォルト)

 → BATプション → 般オプション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ 	E	 マコン 複数ファイルのコンパイル 未使用パブリックを破棄 言語 1 言語 2 コード 最適化 出力 リスト ブリプロt リマークを有効(L(N) 診断を無効(L(S): リマークとして処理(R): 	 2yサ 診断	L荷時設定 MISF 4
デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	•	ワーニングとして処理(W): エラーとして処理(E): ■ すべてのワーニングをエラーとして処理(T)		45-50-7-4711.

9) MISRA-C:2004(デフォルト)

カテゴリ:	工場出荷時	設定
一般オプション	▲ ■ 複数ファイルのコンパイル	
青邦5月4千行 ニュート イトクロナブ	□ 未使用パブリックを破棄	
フンタイム解析		R 1
てんてキキコンハイン		
アセノフラー山市市	 MISRA-C:2004相目はをアクティブ(ご設定) 	
エフリコンハーダ カフタルビルド		
ビルドアカション		
しい(1,7,22,22) リンカ	■ I.1: [required] All code shall conform to ISO 9899:1990 'Programming lan	gl 🔺
デバッガ	1.2: [required] No reliance shall be placed on undefined or unspecified bi	eh
シミュレータ	I.3: [required] Multiple compilers and/or languages shall only be used if	tł
Angel	I.4: [required] The compiler/linker shall be checked to ensure that 31 ch ■ 15 [advisor] Election prior implementations should ensure that 31 ch	nai
CMSIS DAP	1.5: Ladvisory Floating-point implementations should comply with define	30
GDBサーバ	2.1: [required] Assembly language shall be encapsulated and isolated.	
IAR ROMモニタ	2.2. [required] The character sequence (* shall not be used within a con-	
I-jet/JTAGjet		•
J-Link/J-Trace		
TI Stellaris		
Macraigor		
Macraigor PE micro	- OK +**	17

10) MISRA-C:1998 (デフォルト)



11) 追加オプション (デフォルト)

カテコリ: 一般オブション 青師5解析	Î	□
ランタイム解析		リスト プリプロセッサ 診断 MISRA-C:2004 MISRA-C:1998 (追加オプション) 🕢
A ビリンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet	E	



2-1-5. アセンブラ

1) 言語

-

GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace

2) 出力

カテゴリ: 一般オプション ##80-92745		工場出荷時設定
#700/984/J ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet 1 Jink/I-Trace	言語 出力 リスト プリプロセッ ▼デバッグ情報の生成(D)	サ 診断 追加オブション
TI Stellaris Macraigor PE micro	•	ОК * +у>±и

3) リスト

フ テゴリ:		工場出荷時設定
→般オプション 静的解析	<u>^</u>	
ランタイム解析		言語 出力 リスト プリプロセッサ 診断 追加オプション
アセンブラ		<u></u>
出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellarie	E	ッグを含む(H) クロスリファレンスを含む(C) リストをおいし # define #include おたテキスト(T) 内部シンボル(Y) マクロ定義(V) 2行間隔(P) マクロ定義(V) 行散/ページ(G): マクロ実行情報(I) タケ間隔(D): アセンブラ行のみ(S) 後要に応じて設定する
Macraigor		

4) プリプロセッサ

カテゴリ:		
 一般オブション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デバッガ ジミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace 	 言語 出力 リスト グリプロセッキ 「標準のインクルードディレクトリを無 道加インクルードディレクトリ(A): (1行(シンボル定義(D):(1行(こ1シンボル)) 	<u>上場出(可時設定</u>) 診断 追加オブション (こ1ディレクトリ)
TI Stellaris Macraigor PE micro	-	OK twitte

5) 診断

カテゴリ:			工場出荷時設定
 一般オプション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP CMSIS DAP 	E	 言語 出力 リスト プリプロセッサ 診断 追加オプション ワーニング 	
IAR ROM E_A I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	- -	 必要に応じて設定 OK 	定 (キャンセル

6)追加オプション (デフォルト)

カテゴリ: 一般オプション #860-87545	•		工場出荷時設定
■P=198411 ランタイム解析 C/C++コンパイラ		言語 出力 リスト ブリブロセッサ 診断 追加オブション	
アセンブラ		コマンドラインオプションの使用(U)	
出力コンバータ		ーー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel	E		*
CMSIS DAP GDBサーバ IAR ROMモニタ			
I-jet/JTAGjet J-Link/J-Trace			Ŧ
TI Stellaris			



2-1-6. 出力コンバータ







2-1-7. カスタムビルド

1) カスタムツール設定 (デフォルト)

カテゴリ:			
 一般オプション 静的解析 ランタイム解析 	•		
C/C++コンパイラ アセンブラ 出力コンバータ		ファイル名拡張子(F):	
カスタムビルド ビルドアクション		באַעדב (0):	
リンカ デバッガ		出力ファイル(1行に1ファイル)(0):	*
Angel CMSIS DAP		追加入力ファイル(1行に1ファイル)(A):	~
GDBサーバ IAR ROMモニタ			*
I-jet/JTAGjet J-Link/J-Trace TI Stellaris		■ 他のすべてのツールより先にこのツールを実行する(Ψ.
Macraigor	_		

- 2-1-8. ビルドアクション
 - 1) ビルドアクションの設定 (デフォルト)

カテゴリ:	Γ		
一般オプション	^		
静的解析			
ランタイム解析		ドルドアカション語学	
C/C++コンパイラ		CMTYODADAXE	
アセンブラ		プリビルドコマンドライン(P):	
出力コンバータ			
カスタムビルド		ポストビルドコマンドライン(0):	
ビルドアクション	Ξ		
リンカ			
デバッガ			
シミュレータ			
Angel			
CMSIS DAP			
GDBサーバ			
IAR ROMモニタ			
I-jet/JTAGjet			
J-Link/J-Trace			
TI Stellaris			
Macraigor			



2-1-9. リンカ







リンカ設定ファ	イルエディタ	
ベクタテーブル	メモリ領域 スタック/ヒー	-7 412
CSTACK	0×2000]
SVC_STACK	0×100	
IRQ_STACK	0×2000	
FIQ_STACK	0×100	
UND_STACK	0×100	
ABT_STACK	0×100	
НЕАР	0×100	
L		
		1本1チ キャンビル

【CSTACK】スタックサイズ 0x8000 【SVC_STACK】SVCスタックサイズ 0x100 【IRQ_STACK】IRQスタックサイズ 0x100 【FIQ_STACK】FIQスタックサイズ 0x100 【UND_STACK】FIQスタックサイズ 0x100 【UND_STACK】UNDスタックサイズ 0x100 【ABT_STACK】ABTスタックサイズ 0x100 【HEAP】ヒープサイズ 0x100 】

【RZ_T1_init.icf】 ロケーション&セクション定義

/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="\$TOOLKIT_DIR\$¥config¥ide¥IcfEditor¥a_v1_0.xml" */
/*-Specials-*/
define symbolICFEDIT_intvec_start=0x00000000;
/*-Memory Regions-*/
define symbolICFEDIT_region_ROM_start=0x00000000;
define symbolICFEDIT_region_ROM_end = 0x0004FFFF; エディタでの変更不可
define symbolICFEDIT_region_RAM_start_=0x20010000;
define symbolICFEDIT_region_RAM_end = 0x2002FFFF;
/*-Sizes-*/
define symbolICFEDIT_size_cstack = 0x2000;
define symbolICFEDIT_size_svcstack=0x100;
define symbolICFEDIT_size_irqstack=0x2000;
define symbolICFEDIT_size_fiqstack=0x100;
define symbolICFEDIT_size_undstack=0x100;
define symbolICFEDIT_size_abtstack = 0x100;
define symbolICFEDIT_size_heap = 0x100;
/**** End of ICF editor section. ###ICF###*/
define memory mem with size = $4G$;
define region ROM_region= mem:[fromICFEDIT_region_ROM_start to
_ICFEDIT_region_ROM_end_];
define region RAM_region= mem:[fromICFEDIT_region_RAM_start to
_ICFEDIT_region_RAM_end_];

```
define symbol ICFEDIT region STACK start = 0x00050000;
define region EHCI_MEM1_region = mem: [from 0x00060000 to 0x000603FF];
define region EHCI MEM2 region = mem: [from 0x00060400 to 0x00079FFF];
define region OHCI MEM1 region = mem: [from 0x0007A000 to 0x0007A0FF];
define region OHCI MEM2 region = mem: [from 0x0007A100 to 0x0007FFFF];
define symbol region USER PRG start = 0x00000100;
define symbol region USER PRG end = ICFEDIT region ROM end ;
define
        region
                USER PRG region
                                        = mem:[from
                                                        region USER PRG start
                                                                                             to
region USER PRG end ;
define block CSTACK with alignment = 8, size = _ICFEDIT_size_cstack_ {};
define block SVC_STACK with alignment = 8, size = __ICFEDIT_size_svcstack__{};
define block IRQ_STACK with alignment = 8, size = __ICFEDIT_size_irqstack__{;;
define block FIQ_STACK with alignment = 8, size = __ICFEDIT_size_fiqstack__{};
define block UND STACK with alignment = 8. size = ICFEDIT size undstack {};
define block ABT STACK with alignment = 8, size = ICFEDIT size abtstack {};
define block HEAP with alignment = 8, size = ICFEDIT size heap \{\};
initialize by copy {
 readwrite
};
do not initialize {
 section .noinit,
 section.bss
};
do not initialize {
 section EHCI PFL
 section EHCI QH.
 section EHCI_QTD,
 section EHCI_ITD,
 section EHCI SITD,
 section OHCI HCCA,
 section OHCI TD,
 section OHCI ED
};
place at address mem: ICFEDIT_intvec_start {
 section VECTOR_TABLE,
};
place at address mem: region_USER_PRG_start_{
 section d_ldr_prg,
 ro code,
 ro data,
};
```



```
place in EHCI_MEM1_region {
 section EHCI_PFL
};
place in EHCI_MEM2_region {
 section EHCI_QH,
 section EHCI_QTD,
 section EHCI_ITD,
 section EHCI_SITD
};
place in OHCI_MEM1_region {
 section OHCI_HCCA
};
place in OHCI_MEM2_region {
 section OHCI_TD,
 section OHCI_ED
};
place at address mem:_ICFEDIT_region_STACK_start__{
 block FIQ_STACK,
 block IRQ_STACK,
 block SVC_STACK,
 block ABT_STACK,
 block UND_STACK,
 block CSTACK,
 block HEAP,
};
place at address mem:_ICFEDIT_region_RAM_start__{
 rw data,
 section.bss,
 zi
};
```

2) ライブラリ



3)入力 (デフォルト)

	設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト #define 診断 🕙
E	シンボルをキープ(K):(1行に1個)
	ー ローバイナリイメージ(R) ファイル(F): シンボル(S): セクション(E): アラインメント(A):
	E

4) 最適化 (デフォルト)

 一般オブション 一般オブション 静約解析 ランタイム解析 C/C++コンパイラ アセンブラ 出カコンバータ カスタムビルド ビルドアクション リンカ デバッガ ジミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris 		設定 ライブラリ 入力 最適() アドバンスト 出力 リスト #define 診断・ * 小さいルーチンのインライン化と 重複セクションのマージ(M) C++仮想関数除去を実行(P) VFE情報を持たないモジュールがある場合(V)
--	--	---

5) アドバンスト (デフォルト)

カテゴリ:	工場出行	前時設定
→娘オノンヨン 静的解析		
ランタイム解析		=044
C/C++コンパイラ	認定 フ1 ノフツ 八/J 玻迪IE (17 レスレ 出/J リスト #define	19861
アセンブラ		
出力コンバータ		
カスタムビルド	常に含める	
ビルドアクション	■ スタックの使用量解析を有効にする(
リンカ	制御ファイル	
デバッガ	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	
ジミュレータ		
Angel		
CMSIS DAP		
Liet/ITACiet		
1-link/1-Trace		
TLStellaris		
Macraigor		
PE micro		الاصليد لأسراحه

6) 出力



7) リスト

	設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト #define 診断 <td< th=""></td<>
--	--

8) #define (デフォルト)

カテゴリ: 一般オプション 昔船り解析 =>:->->->->->->->->->->->->->->->->->->-		工場出荷時設定
ノノダイム時#1/T C/C++コンパイラ		設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト (#define) 診断・ ♪
アヤンブラ		シンボル定義(D)(1行に1個)
出力コンバータ		
カスタムビルド		
ビルドアクション	=	
リンカ		
デバッガ		
シミュレータ		
Angel		
CMSIS DAP		
GDBサーバ		
IAR ROMモニタ		
I-jet/JTAGjet		· · · · · · · · · · · · · · · · · · ·
J-Link/J-Trace		
TI Stellaris		[
Macraigor		

9)診断 (デフォルト)

ま診断を無効化(D): リマークとして処理(R): ワーニングとして処理(W): エラーとして処理(E): マオズのワーニングをエラーとして処理(T)

10) チェックサム (デフォルト)

カテゴリ:				[工場出荷時設定
一般オプション					
静的解析					
ランタイム解析		入力 最適化 アドバン	スト 出力 リスト	#define i診断	チェックサム 追・
C/C++コンパイラ					
アセンブラ		■ 未使用コードメモリをフ	イルする(F)		
出力コンバータ		フィルパターン	0×FF		
カスタムビルド		開始アドレス(T):	0×0	終了アドレス(E):	0×0
ビルドアクション	=		G)		
リンカ		サイズ(Z):	2/57 ト 🚽	アラインメント(A):	1
デバッガ		77.0 - 10 - 11		0.11001	
シミュレータ				- UX11021	
Angel		フルサイン	ス(*の結果(U)	- 初期/值(I):	
CMSIS DAP		補数(C):	そのまま使用	- 0×0	
GDBサーバ		ビット順(B):	MSBが先頭	 	.了使用(N)
IAR ROME_9		語句内でバ	イトオーダを逆順にする	(R)	o coordio o
I-jet/JTAGjet		チェックサム単位	サイズ(日): 8-hit	-	
J-Link/J-Trace					
TI Stellaris					

11)追加オプション(デフォルト)

1	^								出荷時調	設定
静的解析										
ランタイム解析		マドバンフト	山力	117.5	#define	三会份 后	チェックサル	「自力のオプシン		4
C/C++コンパイラ		TTUAL		221.	#define	10,201				
アセンブラ		ערדב 📃	ラインオブ	ションのほ	更用(U)					
出力コンバータ		1778	ラインオブ	ິນສຽ(G):	(18 5 0-155	75/8)				
カスタムビルド			- 12-02-	22200		1217				
ビルドアクション	=									^
リンカ										
デバッガ										
シミュレータ										
Angel										
CMSIS DAP										
GDBサーバ										
IAR ROMモニタ										
I-jet/JTAGjet										$\overline{\mathbf{v}}$
J-Link/J-Trace										
TI Stellaris										
Macraigor										

2-1-10. デバッガ

1) 設定

ノード"Sample_BARE_R4F"のオ	プション
カテゴリ: - 般オブション 青純的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出カコンパータ カスタムビルド ビルドアクション リンカ <u>デバッガ</u> シミュレータ	
Anger CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	デバイ2起述ファイル(C) 「デフォルトのオーバライド(O) \$TOOLKIT_DIR\$¥CONFIG¥debugger¥Renesas¥R7S910018_R4F.ddf … OK キャンセル
	\$TOOLKIT DIR\$¥CONFIG¥dehugger¥Renesas¥R7S910018 R4Fddf

2) ダウンロード

カテゴリ: 一般オプション			工場出荷時設定
前町9時4所 ランタイム解4所 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ ジミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet	設定 ダウン 二 フラヴ3 二 一 一 一 、 (リーベリファ う デ 一 、 (リーベリファ) 、 、 、 、 、 、 、 、 、 、 、 、 、	ウンロード イメージ 追加オブション マルチコア ブラ マイする(V) ロードしない(XS) シュローダを使用する(U) *フォルトの boardファイルのオーバライド(O) \$TOOLKIT_DIR\$¥config¥flashloader¥ 編集	じんし …
J-Link/J-Trace TI Stellaris Macraigor			

3) イメージ (デフォルト)

一般オノンヨン		INX YE
吉备自约备杂末斤		
ランタイム解析		
C/C++コンパイラ		
アセンブラ	□ 追加イメージのダウンロード	
出力コンバータ	R2:	
カスタムビルド	オフセット: ロデバッグ情報のみ	
ビルドアクション		
リンカ	□ 追加イメージのダウンロード	
テハッガ	パス: …	
ジミュレーダ	オフセット: デバッグ情報のみ	
	□ 追加イメージのダウンロード	
IAR ROM于一久	//Z; [
I-jet/JTAGjet	オフセット: デバッグ情報のみ	
J-Link/J-Trace		
TI Stellaris		

4)追加オプション

יעבינג:			工場出荷時設定
一般オプション	<u>^</u>		
自我の対象性が行うになった。			
C/C++コンパイラ		設定 ダウンロード イメージ (追加オブション)マルチコア ブラ	ガイン
アセンブラ		▼コマンドラインオブションの使用(U)	
出力コンバータ		コマンドラインオブション(C)(1/=(-1コマンド)	
カスタムビルド			
ビルドアクション	E		^
リンカ			
デバッガ			
シミュレータ			
Angel CMSTS DAD			
GDB# - /			
IAR ROMT-2			
I-jet/JTAGjet			~
J-Link/J-Trace			
TI Stellaris		/	
Macraigor			
PE micro	Ψ.		キャンセル

5) マルチコア(デフォルト)

ション・			工場出荷時設定
ー axクランヨン 静的解析	Â.		
ランタイム解析	=_		
C/C++コンパイラ	=×	定 ダウノロート カメーン 超加オフンヨノ くろ	ערעכען אביאי
アセンブラ		-対称型マルチコア(Y)	
出力コンバータ		コアの数(N): 1	
カスタムビルド		非対称型マルチコア(A)	
ビルドアクション	=	□ マルチコアマ2カーチードの有効化(F)	
リンカー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		ポート(P): 53461	
21200 212511-10			
		スレーフワークスペース(W):	
CMSIS DAP		スレーブプロジェクト(S):	
GDBサーバ		スレーブ構成(C):	
IAR ROMELS		■ フレーゴを実行中のターゲット(ロスタッチ	
I-jet/JTAGjet			
J-Link/J-Trace			
TI Stellaris			
Macraigor			

6) プラグイン(デフォルト)

カテゴリ:				工場出荷時設定
一般オプション				
青鲌的解析				
ランタイム解析 C/C++コンパイラ		設定 ダウンロー	ド イメージ 追加オブション マルチコア (プラヴイン)	
アセンブラ		ロードするプラグイ	ンの選択:	
出力コンバータ		HCC-Ware		
カスタムビルド			ce Exporter	E
ドルドアクション	=	CMX		
リンカ	-			
デバッガ		SEGGEB em	60S	
シミュレータ		MOV	500	-
Angel CMSTS DAP		記印月 :	HCC Middleware plugin - has detailed views on of HCC TCPIP stack and eTaskSync scheduler.	internals
		場所:	C:¥Program Files¥IAR Systems¥Embedded Work	bench 7.4¥arm¥p
IAR ROMELS		作成者:	HCC-Embedded	
I-jet/JTAGjet		バージョン	1.0.0.9	
J-Link/J-Trace			1	
TI Stellaris				
Macraigor				
PE micro	-		OK	キャンカル



2-1-1 1. I-jet/JTAGjet

1) 設定

カテコリ: 一般オゴミット)	工場出简時設定
静的解析	
ランタイム解析	
C/C++コンパイラ	
アセンブラ	JESPICE
出力コンバータ	
カスタムビルド	期間(D) 300 ms 遅延(後)(E): 200 ms
ビルドアクション 📲	
リンカ	
デバッガ	フローフ選択プロンプトを常に 表示
シミュレータ	◎ デバッズ後もオンにする(L) シリアルNo・
Angel	● デバック 後にオフにする(S)
CMSIS DAP	
GDBT -/ \	通信ログ(\$)
IAR ROME_9	\$PROJ_DIR\$¥cspycomm.log
I-jet/JTAGjet	
TT Stellaria	
Macrainor	
PE micro	

2) インタフェース

ンタフェース JTAG(J) SWD(S) cJTAG(C) タフェース速度(F) 動検出 ・	明示的なプローブ □ 複数ターゲット ターゲットNo.(□ 複数のCP ターゲット □ JTAGスキ 先行ビッ	設定 デバッグシステム FAPまたはマルチド Uを持つターゲット ・上のCPU No: ャンチェーンに非A ト(P):	ドロップID)(T):0]
	レタフェース速度(F) 自動検出 ▼	y クyı−入速度(F) 自動検出 ▼ 日本 5 0 yi □ JTAGスキ 先行どッ	<u>クタフェース速度(F)</u> 自動検出 ▼ 先行ビット(P):	y クyr ー入速度(F) 自動検出 ▼ □ JTAGスキャンチェーンに非ARM デバイスを含 先行ビット(P): 0

3) Trase (デフォルト)

コテゴリ:				工場出荷時設定
→般オプション #36527#5	<u>^</u>			
時時時時年1月 ラン・カイノー番羽また		\frown		
ノンショム時期1月 - CICエエコトルペイニ		設定 インタフェース Trace	ブレークポイント	
アヤップラ		- トレースデータの収集		
, 出力コンバータ		モード: 自動	▼ >FTBの詳する	т
カスタムビルド				,
ビルドアクション	=	ハッファ制限: 8 サンブ	μ 🔻	
リンカ				
デバッガ		ว่อหวม	クロック設定	
シミュレータ		◉ 自動(A)	CPUクロック(C):	MHz
Angel		🔘 マンチェスタ(M)	SWOプリスケーラ(S): 自動	*
CMSIS DAP		O UART(U)		
GDBサーバ				
IAR ROMモニタ		📃 TraceD0ピンのSWO(W)	
1-jet/JTAGjet				
J-LINK/J-Trace				
Macraicor				
macraigor				

4) ブレークポイント (デフォルト)

ムン・コン・イト カスタムビルド ビルドアグション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/TJAGjet J-Link/J-Trace	 ● 自動(A) ● 八ードウェア(H) ● ソフトウェアブレークポイント復元位置(R) ● ハードウェア(H) ● ソフトウェアブレークポイント復元位置(R) □ パードウェア(W) ● ペリントウェア(W) ● ペリントウェア(W) ● ペリントウェア(W) ● ペリントウェア(W) ● ペリントウェア(W) ● ペロントロント(P) ● ペロントロント(P) ● ペロントロント(P) ● ペロントロント(P) ● ペロントロント(P) ● ペロント(P) ● ペロント(P)<th></th>	
---	--	--

2-2. コア【M3】側の確認

D 🗳 🖬 🕼 🕼 🖌 🖬	B 10 01	
リークスペース	× main.c main s.c	f0 🔹
Sample_BARE_M3 - Debug	• 1 //**	*****
ファイル Sample_BARE G Sample_BARE_W3 - Contput Sample_BARE_R4F B Sample_BARE_R4F D src D output	BR 2 /// 3 /// 4 /// 4 /// オプション(0) メイク(M) コンパイル(C) すべてを再ビルド(B) クリーン(L) C-STAT静的解析(C) ビルドを停止(S) 追加(A)	RZ/T1グループ <(役割> サンプル(main) タブ> I)「Sample_BARE_M3」選択 ②右クリック ③「オプション」選択 意事 r_cpg.c/r_ecm.cか使用している。処理後は利美 ************************************
	削除(V) 名前の変更	define
	ハーション目荘システム(Y) ファイルの場所を開く ファイルのプロパティ(P)	インポート宣言
	アクティブに設定(E)	スタティック関数宣言
一覧 Sample_BARE_M3 Sar ログ Mon May 29, 2017 18:	mple_BAKE_K4F m 30:13: I-jet/JTAGjetドライバを	ロードしています
×	m	



2-2-1. 一般オプション

1) ターゲット

ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デパッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ Iset/JTAGjet J-Link/J-Trace	E	ターゲット 出力 ライブ: プロセッサ選択 ③ コア(O) Co ③ デバイス(D) Re エンディアンモード ④ リトル(L) ● ピッグ(B) ● BE32(3) ④ BE8(8)	 ジ設定 ライブラリオブシ rtex-M3 nesas R7S910018_M3 デ発動小数点演算の話 FPU(F) 無 Dレジスタ(R) - Advanced SIMD (■ン MISRA-C:2004 ② ② ② 数定 ■ いEON)	MISRA-C: 1998
---	---	--	--	--	---------------

2) 出力

	カテゴリ: 	ターゲット 出力 ライブラリ設定 ライブラリオブション MISRA-C:2004 MISRA-C:199 出力ファイル(O) 東行可能ファイル(E) フィブラリ(L) ・ フィブラリ(L) シオブラリ(L) 出力ディレクトリ Debug¥Exe 力ジックトファイル(B): Debug¥Obj Debug¥Obj リストファイル(S): Debug¥List Debug¥List Debug¥List
--	-----------	---

3) ライブラリ設定

サテゴリ: 一般オプション 静地的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーパ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris	E	ターゲット 出力 ライブラリ ライブラリ(L): ノーマル 設定ファイル(O): 参TOOLKIT_DIR\$¥INC¥c*DL ライブラリのスレッドサポート3 低レベルトインタフェースのライフ のなし(N) してまれスティング(S) してARブレークポイント(1)	変更 ライブラリオブション N に/O++ランタイムライブラリの込 ケールインタフェースなし、OE トなし、printf/scanfでのマル 16)道勢が予動小数点数なし。 ib_Config_Normalh を有効にする(E) プラリ実装(B) Stdout/stderr ● セミホスティング経由 ● SWO経由	IISRA-C:2004 MISRA-C:199
Macraigor PE micro	-			0K ***`.d711.

4) ライブラリオプション

カテコリ: 一般オブション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellarie	ターゲット 出力 ライブラリ設定 ライブラリオブション MISRA-C:2004 MISRA-C:19 Printfフォーマッタ ・
--	---

5) MISRA-C:2004 (デフォルト)

アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デパッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ	■ MISRA-Cを有効にする(E) ● MISRA-C:2004 ■ MISRA-C設定のログ(L) ● MISRA-C:1998 MISRA-C:2004規則をアクティブに設定 ■ なし(N) 必須(R) すべて(A) ■ 1.1: [required] All code shall conform to ISO 9899:1990 'Programming langut ■ 1.2: [required] No reliance shall be placed on undefined or unspecified beh ■ 1.3: [required] Multiple compilers and/or languages shall only be used if th ■ 1.4: [required] Multiple compilers shall be checked to ensure that 31 char ■ 1.5: [advisory] Floating-point implementations should comply with defined ■ 2.1: [required] Source shall be encapsulated and isolated.
I-jet/JTAGjet J-Link/J-Trace	۲

6) MISRA-C:1998 (デフォルト)

カテゴリ: 一般オプション 青毛い解析	•						
ランタイム解析 - C/C++コンパイラ		ターゲット	出力	ライブラリ設定	ライブラリオプション	MISRA-C:2004	MISRA-C:19
出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ Ljet/JTAGjet Hink/JTAGjet	ш	MISRA 2 2. [ac 3. [ac 4. [ac 7. [re 7. [re 4. [ac	-C: 1998 U(N) dvisory] dvisory] dvisory] quired] quired] muired] []]	3.規則をアクティブ 必須(F All code shall o Code written ir Assembly lang Provisions sho Only those char Values of char. Trigranhs shall	に設定 ② すべて(A conform to ISO 989 n languages other t uage functions tha uld be made for ap racters and escape acter types shall be not be used	9 standard C, with han C should only t are called from 1 propriate run-time a sequences which e restricted to a c	h no exten A / be used C should b e checking h are defir defined and *
TI Stellaris Macraigor	_						



2-2-2. 静的解析

1) C-STAT 静的解析(デフォルト)

+ ≁u.	_		
ルナコリ: 			
→ 根気オノンヨン 青銅内画型材丘	- Âl l		
ランタイム解析		C-STAT書約的解析 追加オプション	
アセンブラ			
出力コンバータ		0-31817199を進択(3)	
カスタムビルド		設定をインボート(1)…	
ビルドアクション	=	設定をエカフポート(F)	
リンカ ゴバッザ			
シミュレータ		Enable parallel analysis:	2 processes
Angel CMSIS DAP		✓ Enable module timeout:	600 seconds
GDBサーバ		Enable false-positives analysis	
IAR ROMモニタ		✓ Limit messages per check and file:	100 messages
I-jet/JTAGjet			
J-LINK/J-Trace			フィービンへがめりみと70%
11 Stellaris		l	

2)追加オプション (デフォルト)

カテゴリ: ・ MD ナ ポ シューン -			
一般なイノンヨン	Â		
ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ		C-STAT書始的解析 (追加オブショ)	-10
カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ	Ε		
I-jet/JTAGjet J-Link/J-Trace			*

2-2-3. ランタイム解析

1) C-RUN ランタイム解析 (デフォルト)





2-2-4. C/C++コンパイラ

1) 言語1

i.

カテゴリ: 一般オプション 青釉5)解析 ランタイム解析 C/C++コンパイラ		■ 複数ファイルのコンパイル 未使用パブリックを破棄 言語 1 言語 2 コード 最適化 !	工場出荷時設定 出力 リスト ブリブロセッサ 診断 MISF ・
アセンフラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace コンドパック	E	 ● 品(1) ● C ● C++	

2) 言語2

カナコリ: 一般オプション 静的解析 ニンカイノ 約45		■ 複数ファイルのコンパイル ■ 未使用パブリックを破棄	工場出荷時設定
C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	E	言語 1 言語 2 コード 最適化 出力 リスト ブ *char*の型(P) 符号付き ※符号付き 符号付き ※前小数点数動作(F) ● 厳密な適合 緩和(サイズ縮小/高速化) マルチバイト文字サポートを有効にする(E)	リプロセッサ i含単斤 MISF <u>・</u>

3) コード (デフォルト)

カテゴリ:		工場出荷時設定
一般オプション		□ 複数ファイルのコンパイル
青 争的解释析		□ 未使用パブリックを破棄
ランタイム解析		「言語 1」言語 2 コード 最適化 出力 リスト プリプロセッサ 診断 MISE 4
C/C++コンパイラ		
アセンブラ		
出力コンバータ		インタワークコードを生成(D)
カスタムビルド		ー プロセッサモード
ビルドアクション	=	$\bigcirc Arm(A)$
リンカ		Thumb(T)
デバッガ		S manually
シミュレータ		
Angel		位置独立コード/データー
CMSIS DAP		📃 コードおよびリードオンリのデータ (ropi)(C)
GDBサーバ		□ リード/ライトデータ (rwpi)(R)
TAR ROM于一久		動的なリード/ライト初期化しなし(N)
I-iet/ITAGiet		- コードメモリ内のデータRead不生成(N)
1-Link/1-Trace		
TI Stellarie		
Magnicer		
Macraigor		

4) 最適化



5) 出力

→ 取オブション = 一般オブション 自動的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンパータ カスタムビルド ビルドアクション リンカ デパッガ シミュレータ Angel CMSIS DAP GDBサーパ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	 【複数ファイルのコンパイル 未使用パブリックを破棄 言語 1 言語 2 コード 最適化 デバッグ情報の生成(D) codeセクジョン名(C): text .text 	<u>工場出荷時設定</u> <u>出力</u> リスト プリプロセッサ 診断 MISF ・
--	--	--

6) リスト

カテゴリ: 一般オプション 素語の句音245	▲ 【 複数ファイルのコンパイル 】 二場出荷	時設定
ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	■ I 言語 2 コード 最適化 出力 リスト フリプロセッサ 診断 リストファイルの出力(0) ○ アセンブラニーモニック(A) ○ ジザ(D) ■ アセンブラファイルの出力(U) ○ ソースのインクルード(N) ■ 呼出しフレーム情報のインクルード(N)	
Macraigor PE micro	- OK *	ャンセル

7) プリプロセッサ

一般オブション 静的解析 ランタイム解析	 ▲ 複数ファイルのコンパイル ■未使用パブリックを破棄 ■ 言語 1 言語 2 コード 最適化 出力 リスト 	工場出荷時設定 リプロセッサ 診断 MISF ・
マセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris	■標準のインクルードディレクトリを無視() 追加インクルードディレクトリ(A)((1行に1ディレクトリ) \$PROJ_DIR\$/src/Renesas/RZT1_RIN_Engine/Includ \$PROJ_DIR\$/src/Renesas/OMSIS/Include \$PROJ_DIR\$/src/Renesas/Include \$PROJ_DIR\$/src/Renesas/Include \$PROJ_DIR\$/src/Renesas/Include \$PROJ_INC	theck 外す je 「 「 「 「 「 」 「 」 「 」 「 」 」 「 」 「 」 」 「 」 」 「 」 」 「 」 」 「 」 」 」 「 」 」 」 「 」 」 」 」 「 」 」 」 」 」 「 」
Macraigor PE micro		OK キャンセル
	\$PROJ_DIR\$/src/common/inc \$PROJ_DIR\$/src/Renesas/RZT1_RI \$PROJ_DIR\$/src/Renesas/CMSIS/II	N_Engine/Include nclude

8)診断 (デフォルト)

– l> Samble_RAKE_	M3 ()/1.	7532		l
カテゴリ: 一般オプション ま99596245	-	■ 複数ファイルのコンパイル		出荷時設定
ますの 3年70 / ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド		 □ 未成用ハノリッジを破棄 言語 1 言語 2 □ - ド 最適化 出力 リスト □ リマークを有効化(N) 診断を無効化(S): 	ブリブロセッサ 診断	MISF
リンカ デバッガ シミュレータ Angel	Ξ	リマークとして処理(R): ワーニングとして処理(W): エラーとして処理(E):		
CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace		■ すべてのワーニングをエラーとして処理(T)		
TI Stellaris Macraigor PE micro	-			たいけれ

9) MISRA-C:2004 (デフォルト)



10) MISRA-C:1998 (デフォルト)



11) 追加オプション (デフォルト)





2-2-5. アセンブラ

1) 言語

-

CMSIS DAP GDBサーバ IAR ROMモニタ I-jt/JTAGjet

2) 出力

カテゴリ:		工場出荷時設定
一般オプション	^	
吉胆匀解释析 □> ┗/+ ₽₽+5		
フンダイム時代	言語 出力 リスト プリプロセッサ	診断 追加オプション
マセングラ		
出力コンバータ		
カスタムビルド		
ビルドアクション	=	
リンカ		
デバッガ		
シミュレータ		
Angel		
CMSIS DAP		
I-iet/JTAGiet		
J-Link/J-Trace		
TI Stellaris		
Macraigor		
PE micro	*	

3) リスト

 リストファイルの出力(0) ッグを含む(H) クロスリファレンスを含む(C) リストをお(L) #define #inclose さいたテキスト(T) 内部シンボル(Y)
 マクロ定義() マクロ症義(P) マクロ拡張子(N) マクロ実行情報(0) ワヤセンブラ行のみ(S) (根数行コード(U) (日本のに応じて設定する)

4) プリプロセッサ

カテゴリ: 一般オプション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP	エ場出荷時設定 ゴ島加オブション 電準のインクルードディレクトリを無視(S) Check タトす 追加インクルードディレクトリ(A): (1行に1ディレクトリ) ・ ジンボル定義(D)(1行に1ジンボル) ・
GDBサー/S IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	

5) 診断

カテゴリ: 一般オプション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーパ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace	 ■ 認力 リスト ブリブロセッサ ワーニング ● 有効(E) ● すべてのワー ● 無効(D) ● 特定ワーニ ● ワーニング ■ 最大エラー数(M): 	工場出荷時設定
Macraigor		

6)追加オプション(デフォルト)

カテゴリ: 一般オプション ま99596245						出荷時設定
ランタイム解析 C/C++コンパイラ	言語	出力 リン	(h วีมวีอt	2ッサ 診断	追加オプション	
アセンブラ 出力コンバータ カスタムビルド ビルドアクション	דב	ンドラインオ : ンドラインオフ	グションの使用(グション(C):(1行	נ ו) (נוסדרא)		*
リンカ デバッガ シミュレータ Apgel						
CMSIS DAP GDBサーバ IAR ROMモニタ						
I-jet/JIAGjet J-Link/J-Trace TI Stellaris						Ŧ
Macraigor						



2-2-6. 出力コンバータ

1) 出力





2-2-7. カスタムビルド

1) カスタムツール設定 (デフォルト)

カテゴリ:	[
一般オプション			
静的解析			
ランタイム解析		カスタムツール設定	
C/C++コンパイラ			
アセンブラ		ファイル名拡張子(F):	
出力コンバータ			
カスタムビルド			
ビルドアクション	E		
リンカ		出力ファイル(1行に1ファイル)(0):	
デバッガ			*
シミュレータ			-
Angel			· ·
CMSIS DAP		追加入力ファイル(1行に1ファイル)(A):	
GDBサーバ			~
IAR ROMELS			-
I-jet/JTAGjet		● 他の才がてのいこれ上り生(このいこれを実行する)	*
J-Link/J-Trace		■他のすべてのケールより元にこのケールを美行する	
TI Stellaris			
Macraigor			

2-2-8. ビルドアクション

1) ビルドアクションの設定 (デフォルト)

カテゴリ:			
一般オプション	^		
青鲌的解析			
ランタイム解析		ドルドアクション設定	
C/C++コンパイラ			
アセンブラ		プリビルドコマンドライン(P):	
出力コンバータ			
カスタムビルド		ポストビルドコマンドライン(0):	
ビルドアクション	Ξ		
リンカ			
デバッガ			
シミュレータ			
Angel			
CMSIS DAP			
GDBサーバ			
IAR ROM王二久			
I-iet/JTAGiet			
J-Link/J-Trace			
TI Stellaris			
Macraigor			
nucruigoi	-		



2-2-9. リンカ

1) 設定



【M3_image.icf】 ロケーション&セクション定義



2) ライブラリ

ከታግሀ:		
		工場出荷時設定
あたり シンヨン 吉争的句麗な析		
ランタイム解析		
C/C++コンパイラ	設定 マリフジノ人力 最適化 アドバンスト	ヽ 出力 リスト #define 診断 <u>▲ </u>
アセンブラ	▼ 自動ランタイムライブラリ選択(A)	
出力コンバータ	1月加ライブラリ(L):(1行(21個))	
カスタムビルド		A
ビルドアクション	=	
リンカ		-
デバッガ		(0)
シミュレータ		2 (U)
Angel	● エノドリシノハ(KE) <u></u> lar_brogram_ste	
CMSIS DAP	0 7 7 7 7 9 5 C (2 # 5 (D)	
List/ITACist		
1-jet/JTAGjet		
TI Stellaris		
Macraigor		
PE micro	T	الطردين في

3)入力(デフォルト)

			静的解析 =>.5/1 87#5
#define 診断	出力 リスト #define	設定 ライブラリ	フンダイム時料ケ
		シバッボルをキープ(K	マセンブラ
		55/10/21 5 (8	, こうテラ 出力コンバータ
			カスタムビルド
		=	ビルドアクション
			リンカ
			デバッガ
			シミュレータ
			Angel
			CMSIS DAP
アラインメント(A):	セクション(E): アライン:	ファイル(F):	
			IAR ROME_S
			rjegji Adjet
			1. ink/1. Trace
2	セクション(E): 7	ローバイナリイメー ファイル(F):	Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet

4) 最適化 (デフォルト)

 → 般オプション 静的解析 ⇒ シスクム解析 ⇒ シスクム解析 ⊂ (C++コンパイラ アセンブラ 出カコンバータ カスタムビルド ビルドアクション アセンブラ 出カコンバータ カスタムビルド ビルドアクション アビンブラ 出カロンバータ カスタムビルド ビルドアクション アバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ Lipt(JTAGjet Lipt(JTAGjet 	工場出荷時設定
TI Stellaris Macraigor	

5) アドバンスト (デフォルト)

カテゴリ:	工場出行	前時設定
→娘オノンヨン 静的解析		
ランタイム解析		=044
C/C++コンパイラ	認定 フ1 ノフツ 八/J 玻迪IL (17 レスレ 出/J リスト #define	19861
アセンブラ		
出力コンバータ		
カスタムビルド	常に含める	
ビルドアクション	■ スタックの使用量解析を有効にする(
リンカ	制御ファイル	
デバッガ	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	
ジミュレータ		
Angel		
CMSIS DAP		
Liet/ITACiet		
1-link/1-Trace		
TLStellaris		
Macraigor		
PE micro		العطر لأسرعة

6) 出力



7) リスト

	設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト #define 診断 <td< th=""></td<>
--	--

8) #define (デフォルト)

カテゴリ: 一般オプション 昔船り解析 =>:->->->->->->->->->->->->->->->->->->-		工場出荷時設定
ノノダイム時#1/T C/C++コンパイラ		設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト (#define) 診断・ ●
アヤンブラ		シンボル定義(D)(1行に1個)
出力コンバータ		
カスタムビルド		
ビルドアクション	=	
リンカ		
デバッガ		
シミュレータ		
Angel		
CMSIS DAP		
GDBサーバ		
IAR ROMモニタ		
I-jet/JTAGjet		•
J-Link/J-Trace		
TI Stellaris		[
Macraigor		

9)診断 (デフォルト)

カテゴリ: 一般オプション 静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ ショータ	エ場出荷時設定 ライブラリ 入力 最適化 アドバンスト 出力 リスト #define 診断 チェッ * リマークを有効化(N) 診断を無効化(D): リマークとして処理(R): ワーニングとして処理(W):
Angei CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris	エラーとして処理(E): 「 すべてのワーニングをエラーとして処理(T)

10) チェックサム(デフォルト)

一般オブション 静的解析					上场出的时款定
百尹日刘冉年代广					
二、					
フノダイム時料ケ	入	、力 最適化 アドバン	シスト 出力 リスト	#define 話診世所	チェックサム追・
マセンブラ		■ 未使用コードメモリな	フィルオろ(F)		
ノ ビンノン 出力コンバータ		コイルパターン フィルパターン	0xFF		
カスタムビルド		見始アドレス(T):	Π×Π	終了アドレス(F):	0×0
ビルドアクション	=	「エーシンクサン生式」	(9)	000 1 7 1 12700E74	0.00
リンカ		サイズ(7):	00 00875 -	アラインメント(A):	1
デバッガ			27111 +		
シミュレータ		アルコリズム	CRC16	▼ 0×11021	
Angel		フルサイ	ズでの結果(U)	~初期値(I):	
CMSIS DAP		ネ捕裝如(℃):	そのまま使用	- 0×0	
GDBサーバ		ビット順(B):	MSBが先頭		して使用(N)
IAR ROME_9		- 語句内でパ	イトオーダを逆順にする	5(R)	
1-jet/JTAGjet		チェックサム単位	立サイズ(H): 8-bit	-	
TT Stellarie					
11 Stellaris					

11)追加オプション(デフォルト)

1	^								出荷時調	設定
静的解析										
ランタイム解析		マドバンフト	山力	117.5	#define	三会份 后	チェックサル	「自力のオプシン		4
C/C++コンパイラ		TTUAL		221.	#define	10,201				
アセンブラ		ערדב 📃	ラインオブ	ションのほ	更用(U)					
出力コンバータ		1778	ラインオブ	ິນສຽ(G):	(18 5 0-155	75/8)				
カスタムビルド			- 12-02-	22200		1217				
ビルドアクション	=									^
リンカ										
デバッガ										
シミュレータ										
Angel										
CMSIS DAP										
GDBサーバ										
IAR ROMモニタ										
I-jet/JTAGjet										$\overline{\mathbf{v}}$
J-Link/J-Trace										
TI Stellaris										
Macraigor										

2-2-10. デバッガ

1) 設定

カテゴリ: 一般オプション	^	工場出荷時設定
静的解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムピルド ビルドアクション リンカ デバッガ ジミュレータ Angel CMSIS DAP GDBサーバ	=	後定 ダウンロード イメージ 追加オプション マルチコア ブラヴイン ドライバ(D) 「Fjet/JTAGjet セットアップマクロ(E) マクロファイルの使用(U) デバイ2記述ファイル(C) 「マティルトのオーノジライド(D) 」 「デバイ2記述ファイル(C) 「マティルトのオーノジライド(D) 」
IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	~	\$TOOLKIT_DIR\$¥CONFIG¥debugger¥Renesas¥R7S910018_M3.ddf … OK キャンセル

2) ダウンロード

カテコリ: 一般オプション	<u>^</u>	工場出荷時設定
音野の解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel	ш	 設定 ダウンロード イメージ 追加オブション マルチコア ブラヴイン ダウンロードしない(XS) ブラッシュローダを使用する(U) デフォルトのboardファイルのオーバライド(O) \$TOOLKIT_DIR\$¥config¥flashloader¥ 編集
CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor		

3) イメージ (デフォルト)

一般オノンヨン		INX YE
吉备自约备杂末斤		
ランタイム解析		
C/C++コンパイラ		
アセンブラ	□ 追加イメージのダウンロード	
出力コンバータ	R2:	
カスタムビルド	オフセット: ロデバッグ情報のみ	
ビルドアクション		
リンカ	□ 追加イメージのダウンロード	
テハッガ	パス: …	
ジミュレーダ	オフセット: デバッグ情報のみ	
	□ 追加イメージのダウンロード	
IAR ROM于一久	//Z; [
I-jet/JTAGjet	オフセット: デバッグ情報のみ	
J-Link/J-Trace		
TI Stellaris		

4)追加オプション

יעבינג:			工場出荷時設定
一般オプション	<u>^</u>		
自我の対象性が正			
C/C++コンパイラ		設定 ダウンロード イメージ (追加オブション)マルチコア ブラ	ガイン
アセンブラ		▼コマンドラインオブションの使用(U)	
出力コンバータ		コマンドラインオブション(C)(1/=(-1コマンド)	
カスタムビルド			
ビルドアクション	E		^
リンカ			
デバッガ			
シミュレータ			
Angel CMSTS DAD			
GDB# - /			
IAR ROMT-2			
I-jet/JTAGjet			~
J-Link/J-Trace			
TI Stellaris		/ \\	
Macraigor			
PE micro	Ψ.		キャンセル

5) マルチコア (デフォルト)

ション・			工場出荷時設定
ー axクランヨン 静的解析	Â.		
ランタイム解析	=_		
C/C++コンパイラ	=×	定 ダウノロート カメーン 超加オフンヨノ くろ	ערעכען אביאי
アセンブラ		-対称型マルチコア(Y)	
出力コンバータ		コアの数(N): 1	
カスタムビルド		非対称型マルチコア(A)	
ビルドアクション	=	□ マルチコアマ2カーチードの有効化(F)	
リンカー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		ポート(P): 53461	
213-1-b			
		スレーフワークスペース(W):	
CMSIS DAP		スレーブプロジェクト(S):	
GDBサーバ		スレーブ構成(C):	
IAR ROMELS		■ フレーゴを実行中のターゲット(このタッチ	
I-jet/JTAGjet			
J-Link/J-Trace			
TI Stellaris			
Macraigor			

6) プラグイン(デフォルト)

カテゴリ:				工場出荷時設定
一般オプション				
静的解析				
ランタイム解析 C/C++コンパイラ		設定 ダウンロー	ド イメージ 追加オブション マルチコア (プラヴイン)	
アセンブラ		ロードするプラグイ	ンの選択:	
出力コンバータ		HCC-Ware		
カスタムビルド			ce Exporter	E
ドルドアクション	=	CMX		
リンカ	-			
デバッガ		SEGGEB em	60S	
シミュレータ		MOV	500	-
Angel CMSTS DAP		記 8月:	HCC Middleware plugin - has detailed views on of HCC TCPIP stack and eTaskSync scheduler.	internals
		場所:	C:¥Program Files¥IAR Systems¥Embedded Work	bench 7.4¥arm¥p
IAR ROMELS		作成者:	HCC-Embedded	
I-jet/JTAGjet		バージョン	1.0.0.9	
J-Link/J-Trace			1	
TI Stellaris				
Macraigor				
PE micro	-		OK	キャンカル



2-2-1 1. I-jet/JTAGjet

1) 設定

カテコウ. 一般オプション	工場出荷時設定
静的解析 ランタイム解析 C/C++コンパイラ	設定 インタフェース Trace ブレークポイント
アセンブラ 出力コンバータ カスタムビルド	無効 (Uセットなし) 東別(Uセットなし) 東間(D): 300 ms 遅延(後)(E): 200 ms
リンカ デバッガ シミュレータ Angel	■ - ターゲット電源 - エミュレータ ■ プロープから供給(F) ● デバッグ後もオンにする(L) ● デバッグ後にオフにする(S)
CMSIS DAP GDBサーバ IAR ROMモニタ	□通信ログ(C) \$PROJ DIR\$¥cstycomm.log
J-Link/J-Trace TI Stellaris	
PE micro	- OK キャンセル

2) インタフェース

カテゴリ: 一般オプション 静約解析 ランタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ ジミュレータ Angel CMSIS DAP GDBサーバ	▲ 設定 (1)タフ: フローブ設定 ●自動 つアィルが ●明示的 1ンタフェース ●JTAG(J) ● SWD(S) ● cJTAG(C) ● cJTAG(C)	エ場出荷時設定 Trace ブレークポイント プローブ設定ファイル デフォルトのオーパライド(0) 第PROJ_DIR¥PZT1.ProbeConfig CPU: M3 選択 明示的なプローブ設定 祥観数ターゲットデバッグシステム ターゲットNo(TAPまたはマルチドロップID)(T):0 祥観数のCPUを持つターゲット ターゲット上のCPU No: 0 ア・ゲットンの (の)
IAR ROM <u>E_</u> I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro		JTAGスキャンチェーンに非ARM デバイスを含む 先行ビット(P): 0 OK たかけれ

3) Trase (デフォルト)

コテゴリ:				工場出荷時設定
ー般オプション #365-87#5	<u>^</u>			
朝回い時時代 ランタイム解析				
C/C++コンパイラ		設定 インタフェース Vrace	/ブレークボイント	
アセンブラ		トレースデータの収集		
出力コンバータ		モード: 自動	▼ >ETBの許可	
カスタムビルド		バッファ制限: 8 サンプ		
ビルドアクション	=			
リンカ デバッガ		ว่อเงิวม	っクロック設定	
シミュレータ			CPUクロック(C):	MHz
Angel		マンチェスタ(M)	SW0-117 5-5(S)	
CMSIS DAP		() LIART(1)	30000000000000000000000000000000000000	T
GDBサーバ		<u> </u>		
IAR ROMモニタ		📃 TraceD0ピンのSWO(W	0	
I-jet/JTAGjet				
J-LINK/J-Trace				
Macrainor				
- act algor	-			

4) ブレークポイント (デフォルト)

 一般オブション 一般オブション 「緑台)解析 シッタイム解析 C/C++コンパイラ アセンブラ 出力コンバータ カスタムビルド ビルドアクション リンカ デバッガ シミュレータ Angel CMSIS DAP GDBサーバ IAR ROMモニタ I-jet/JTAGjet J-Link/J-Trace 	設定 1ンタフェース Trace フレークポイント ・ 自動(A) ・ ハードウェア(H) ・ ソフトウェアブレークポイント復元位置(R) ・ こall_main ・ ソフトウェアグレークポイント復元位置(R) ・ こall_main ・ ソフトウェアグレークポイント復元位置(R) ・ こall_main ・ マリロット(T) ビブリフェッチ(P) ・ CORERESET ビ STATERR ・ ジフトウェア(M) ・ マリロット(T) ビブリフェッチ(P) ・ CORERESET ビ STATERR ・ マリロット ・ マーター ・ マリロット ・ マリロット ・ マーター ・ マリロット ・ マーター ・ マリロット ・ マーター ・ マリロット ・ マーター ・ マー ・ マー ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
TI Stellaris Macraigor	

- 3. サンプルプロジェクト「Sample_BARE」をビルドする。
 - 3-1. コア【R4F】 側サンプル「Sample_BARE_R4F」をビルドする。
 - 1) すべてを再ビルド

) 🚅 🖬 🕼 😹 🐰	一番冊 50	📣 🏔 🛤 0 👷 😹 🕭
ークスペース	main_s.c	f0 👻
Debug	1 //***********************************	******
ファイル	B2: 2 // 3 // RZ/T1グループ	
∃ <mark>]</mark> Sample_BARE_R4	オプション(の) (公割)、 みついでは (motion)	
- E Common	(位計) リンフル(main)	
🖃 🗀 monitor	KCPU> RZ/T1 Cortex-R4F Ex	tC(25,00MHz) Iclk(600MHz)
-⊞ <mark>⊡</mark> sample	コンパイル(C) (基板) ()プロジェクトタ	
-± Output	「「人てを用ビルト(目)」「「」」」「」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」	
	Sample_BARE_	_R4F」をクリック
	C-STAT静的解析(C) ・注意 ②右クリック	10.00
	の「ナベアな再ビア	可していた。
	ビルドを停止(5) (3) 「9** (を円し)	
-	追加(A) 🕨 ud e	
	ude Unton	
	2010年(1) 名前の亦重	
	define	
	バージョン管理システム(Y) ・	
	ファイルの場所を開く	
	インホート旦言 ファイルのプロパティ(P)	
	アクティブに設定(E)	
	30 // スタティック関数宣言	
	32 static void ecm_initial(void	1),
		•
ログ		
4	III	•

2) ビルド結果



3-2. コア【M3】側サンプル「Sample_BARE_MB」をビルドする。

1) すべてを再ビルド



2) ビルド結果



- 4. サンプルプロジェクト「Sample_BARE」をデバッグする。
 - 4-1. MP-RZT1-01 のデバッグ準備
 - 1) コア【R4F】 側とコア【M3】 側を Ijet のみ使用してデバッグする場合



2) コア【R4F】 側を DEFnano、コア【M3】 側を Ije を使用してデバッグする場合



4-2. コア【R4F】側のデバッッガ初期起動

4-2-1. I-jet を使用(詳細なデバッグ操作は IAR 提供「デバッグガイド」を参照)

1) サンプル「Sample_BARE_R4F」をダウンロードする。

Debue ファイル 回 ファイル 回 コークスペース クリック コークス クリック コークス クリック コークス クリック コークス クリック コークス クリース コークス アインク コークス アングル ロークス アングル ロークス アングル ロークス アングル ロークス アングル ローク アングル ローク ローク ローク ローク コーク ローク コーク ローク コーク ローク コーク ローク コーク ローク コーク ローク	ノアイル(F) 編集(E) 表示(V) プロジェクト	(P) 1-jet/JIAGjet(1) ツール(1) ワイントワ(W) ヘルフ(H)
Debue 1 // ***********************************		
Debue 1 //***********************************	<u>J-03/K-X</u>	ain_s.c
ファイル 四 ① Sample_BARE_R4F・ グリック ① Sample_BARE_R4F・ グリック ① Sample_Banple グリック ③ Output 〇リック ③ Output 〇リック ⑤ Sample_Banple 〇リック ⑤ Sample_Banple 〇リック ○ Output 〇リトルエンディアン ○ Output 〇ロリトルエンディアン ○ Output 〇リトルエンディアン ○ Output 〇ロートル ○ Output 〇ロートル ○ Output 〇ロートル ○ Output ○ Output ○ Output ○ Output	Debug 🗸	
Sample_BARE_M3 Sample E * * *******************************	ファイル 📴	$\frac{2}{3}$ // RZ/IIJ π - τ $(7 \cup \gamma)$
日日 monitor 6 // < タブ> 4タブ編集 日日 monitor 7 // < CPU> RZ/TI Cortex-R4F ExtC(25,00MHz) Iclk(600MHz) 日日 monitor 8 // < MD> main_s.c 日日 monitor 9 // < EW MD> main_s.c 日日 monitor 9 // < AU MD> main_s.c 日日 monitor 9 // < EW MP-RZTI-01 10 // < Debug> リトルエンディアン 11 // < Debug> リトルエンディアン 12 // < 注意事項 CMT0ユニットは使用に注意が必要!! 14 // r_opg.o/r_ecm.oが使用している。処理後は利用可 15 // ***********************************		5 // <役割> サンプル(main)
H → monitor	- 🖽 🗀 common	6 // 〈タブ〉 4タブ編集
	- 🕀 🧰 monitor	7 // <cpu> RZ/T1 Cortex-R4F ExtC(25,UUMHz) lclk(6UUMHz)</cpu>
Land 10 10 10 11 // (Debug> リトルエンディアン 13 // 注意事項 CMT0ユニットは使用に注意が必要!! 14 // : 15 //: 16 //: 17 #include "main_s.h" 18 #include "Umonitor.h" 19 //: 20 //: 21 // define 22 //: 21 // define 22 //: 10 //: 11 //: 12 //: 13 //: 14 //: 15 //: 16 //: 17 #include "Main_s.h" 18 #include "Umonitor.h" 19 //: 20 //: 21 //: 22 //: 23 //: 24 //: 25 /: 26 /: 27 /: 28 /: 29 /: 20 /: 21 /: 22 /: 23 : 24 : <	└─⊞ 🗋 sample	9 // <基板> MP-RZT1-01
11 // 〈Debug〉 リトルエンディアン 12 // 注意事項 CMT0ユニットは使用に注意が必要!! 13 // 注意事項 CMT0ユニットは使用に注意が必要!! 14 // r_cpg.c/r_ecm.cが使用している。処理後は利用可 15 // #include "main_s.h" 16 // #include "Umonitor.h" 19 // define 20 // define 21 // define 22 // define 23 // define 24 // define 25 // define 26 // define 27 // define 28 // define 29 // define 21 // define 22 // define 23 // define 24 // define 25 // define 27 // define 28 // define 29 // define 21 // define 22 // define 21 // define 22 // define 23 // define 24 // define 25/5/07 EU/F // define	- Output	10 //
12 // 注意事項 CMT0ユニットは使用に注意が必要!! 14 // 「ops.c/r_ecm.oが使用している。処理後は利用可. 15 // ***********************************		11 // <debug> リトルエンディアン</debug>
19 20 21 // define 22 23 // define 22 23 // define 22 23 // define 22 23 // define 22 23 // define 22 23 // ・ 一覧 Sample_BARE_M3 Sample_E () () () () () () () () () () () () ()		 13 // 注意事項 CMIDユニットは使用に注意が必要!! 14 // r_orgs.c/r_ecm.cが使用している。処理後は利用可 15 // 16 //***********************************
20 /// define 21 // define 22 //		19
ビューバー		20 //
-覧 Sample_BARE_M3 Sample_E → 23		22 //
- 覧 Sample_BARE_M3 Sample_E▲ ↓ ↓ ・ ログ Thu Jun 08, 2017 10:34:19: I-jet/JTAGjetドライバをロードしています ↓ … ↓ … ↓ … ↓ … ↓ … ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓		23
ログ Thu Jun 08, 2017 10:34:19: I-jet/JIAGjetドライバをロードしています ・	一覧 Sample_BARE_M3 Sample_E	4
Thu Jun 08, 2017 10:34:19: I-jet/JIAGjetドライバをロードしています		
・ 「	Thu Jun 08, 2017 10:34:19: I-jet,	JIAGjetドライバをロードしています
デバッグログ [ビルド	· · · · · · · · · · · · · · · · · · ·	
	デバッグログ ビルド	

2) デバッガ起動画面

		(A) 1-jet/JTAGjet(I) 9-)	
ᆸᇔᇔᇔᇔᇥ ᆕᇻᇔᇔᇔᇔᇔᇔᇥ ᄧᄦ	×		₩2 E# E : 4: 91: 2> 22 (
フークスペース ×	main_s.c loader_init.asm	f0 👻 🗙	逆アセンブリ
Debug イ ファイル 常語 日本 Sample_BAR イ ー Sample_BAR イ ー Common ー 田 Common ー 田 Common ー 田 Common ー 田 Common	69 : Return Value : noil 70 :************************************	ne ************************************	逆アセンブリ びx00 逆アセンブリ Oxe0: 0x0000 Oxe4: 0x0000 Oxe4: 0x0000 Oxe8: 0x0000 Oxe6: 0x0000 Oxr6: 0x0000 Oxrf0: 0x0000 Ox100: 0xf100 Ox104: 0xe593 cps #18 ; IR0 n Ox108: 0xf100 Idr sp, =SFE(IR(Ox108: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(Ox100: 0xf100 Idr sp, =SFE(IR(
	< <u> </u>	· 4	
Thu Jun 08, 2017 10:37:43	: INFO: Configuring trace using ': : FTR ID:0v00000021 Canacity:4K	SWO,ETB' setting R (AV complex)	

- 4-2-2. DEFnano を使用(詳細なデバッグ操作は Aone 提供「DEFnano_Vx_xx.pdf」を参照)
 - 1) サンプル「Sample_BARE_R4F」をダウンロードする。

ダウンロード(D)	
$\overline{P} \sqrt{7} \overline{\Box} - \overline{F}(U)$	
NU77400	
2 Not (1 = 1) - (V)	
ンンハッレ読込み(*)	②【ノアイル】-【タリンロート】を選択
アブソリュートファイル設定(A	A) -0.00002020
CDU 50 ==== 1 -= (C)	-uxuuuuszaa S pc. lr. ±0x4
CPU設定読込み(S)	=0x00000018
CPU設定登録(R)	=0×000031F0
スクリプト実行(L)	Q r0, r0, r0
オフライン作業(0)	EQ r0,r0,r0
オフライン環境設定(M)	50 r0, r0, r0
約了(F)	20 r0, r0, r0
h000003C+ 0000000	EW rU,rU,rU AMDEO +0 +0
00000040: 00000000	ANDEQ r0,r0,r0
00000044: 00000000	ANDEQ r0,r0,r0
00000048: 00000000	ANDEQ r0,r0,r0
	①Start をクリック
00000054: 00000000	
SB1 000000000	LR ▼ C SB2 000000000 ▼ CLR ▼ 実行回数 1 ソフトフ*
	▼ (なし) ▼ [char ▼ 16進 ▼ ■ 周期
Go Bre RstR4F RstM	3 Win Reg Watch Sym Trace Step Cirac CStep IntFlg

IDEFnano アブソリュート/ヘキサファイル指定	でのダウンロード
CO O ▼ 🖟 « R4F ト Debug ト Exe	 ▼ 4 Exeの検索
整理 ▼ 新しいフォルダー	8= - 🔟 📀
RZT1_Sample_BARE	▲ 名前 更新日E
퉲 МЗ	Sample_BARE_R4F.out 2017/0
🔒 R4F	
Debug	
Le Exe	
🍶 List	
🔒 Obj	
📔 settings	▼ ()
Sample_BARE_R4F.out 状 PSpice Simulation Output 更新日間	兄: 34 共有 侍: 2017/06/07 18:39
ファイル名(N): Sample_	BARE_R4F.out ABSFile(*.x,*.axf,*.out)
	聞く(0) #ヤンセル

<RZT1_Sample_BARE><R4F><Debug><Exe>[Sample_BARE_R4F.out]を選択後、「開く」をクリック

2) デバッガ起動画面

a state of the	ano & S	erialFlashWriter Ver3	00A RZ/T1 AHnano Ver1.00(2016	/11/25)	
ファイル(F) ・	データ([D) 実行(G) ブレーク	y(B) 割り込み(I) オプション(O)	へルプ(H)	
停止 割辺	Unt Pr	2.0~0000000 □ 周期 [View CPUBZ/T1(B7S910018)	Advance25.0000 Tv24 Tr	
	#01.				
	#21. 91		leader initi	or_rodin.asm同迷 _	<u> </u>
00000000	22		b Hodefined Interrupt : S	tart+0000 、りセッド	
00000008:	23	syc bandler:	b syc bandler : 8	tart+0008 : ソフトウェ	ア創け込み
0000000C:	24		b prefetch handler : S	tart+000c : プリフェッ	チアボート
00000010:	25		b Abort_Interrupt ; S	tart+0010 : データアボ	
00000014:	26	reserved_handler:	subs pc, Ir, #4 ; S	tart+0014 : Reserved	
00000018:	27	irg_handler:	b irg_handler ; S	tart+0018 : IRQ	
0000001C:	28		b FigHandler_Interrupt ; S	tart+001c : FIQ(NMI)	
00000020:	29	;			
	30	END			
	31	; End of File			
SB1 0000000	000	CLR	✓ C SB2 000000000	LR 実	行回数 1
5 2B1 0000000	000	CLR	▼ C SB2 00000000 ▼ (なし) ▼ [char	▼CLR ▼ 実	行回数 1 ソフトフ・
5 SB1 [0000000	000	CLR	▼ C SB2 00000000 ▼(なし) ▼[char	▼CLR ▼ 実	行回数 1 ソフトフィー
5 SB1 0000000	000 -	(CLR .R4F RstM3 ₩in	▼ C SB2 000000000 ▼(なし) ▼ [char Reg Watch Sym T	▼CLR ▼ 実 ▼ 16進 ▼ race Step CTra	行回数 1 「周期 」 」C CStepIntFlg
5 SB1 0000000 Go Brea 8_sys =0x0000	000 - ak Rst 00000 R	CLR 		▼CLR ▼ 実 ▼ 16進 ▼ race Step CTra 00 R12_sys=0x00000000	行回数 1 <u>ソフトフ^cI 周期</u> .c CStep IntFIg SP_sys =0x00000000 LR_sys =0x00

4-3. コア【M3】側のデバッッガ初期起動

4-3-1. コア【M3】 側のリセット解除

RZ/T1 の電源投入後、および RESET#端子リセット時には、Cortex-M3 側は初期状態としてリセット状態です。コア【M3】側のデバッグを開始させるには、コア【R4F】側から、ソウトウェアリセットレジスタ2(SWRR2)を制御して、リセット解除する必要があります。

void cn	n3_start(void)
t	R_RST_WriteEnable();
	SYSTEM.SWRR2.LONG = 0x4321A50F; // Reset asm("isb"); // Ensuring instruction-changing
	SYSTEM.SWRR2.LONG=0x0; // Reset 解除 asm("isb"); // Ensuring instruction-changing
}	R_RST_WriteDisable();

1) Ijetを使用して M3 側のリセット解除(R4F 側のデバッガを起動)

る Sample_BARE - IAR Embedded Workbench IDI ファイル(F) 編集(E) 表示(V) プロジェクト(P) ヘルプ(H)	E - ARM 7.60.2 デバッグ(D) 逆アセンブル(A) I-jet/JTAGjet(I) ツール(T) ウィンドウ(W)	
□ こ 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	main_s.c loader_init.asm main_s.c	₫₽> ₫₽> n0 - x
ファイル 常語 時端 つ Sample_BARE_R4F - Debug ・ 一日 Src 一日 common 一日 common 二日 c	<pre></pre>	
× ログ ▲ Thu Jun 08, 2017 13:34:42: デバッグセ Thu Jun 08, 2017 13:35:51: ヒットした	セッションの初期化中に1件のワーニングがありました。 たブレークポイント: コード @ main_s.c:61.5、タイプ:デフォルト(自動)	• •
ะ		×
准備完了	(규 61 제 5	

2) DEFnano を使用して M3 側のリセット解除

🔤 Aone DEFnano & Serial	lashWriter Ver3.00A RZ/T1 AHnano Ver1.00(2016/11/25)
ファイル(F) データ(D)	¥行(G) ブレーク(B) 割り込み(I) オプション(O) ヘルプ(H)
停止 割込 DI PC 0×0	000000 [] 周期 [] View CPURZ/T1(R7S910018) Advance25.0000 Ix24 TOP 0x00000000 SIZ 0x00 😱
00000000 📮	▼ CSrc CMix ⊙Asm ▼ 高速 ▼
00000000: EA00003E	B =0x00000100
• 00000004: EA000C8E	B =0×00003244
• UUUUUUU8: EAFFFFE	B =0x0000008
• 00000000: EA003845	D -0X00026026 B =0√00003258
• 00000014: E25EE004	SUBS pc. (r. ±0x4
• 00000018: EAFFFFE	B =0x00000018
• 0000001C: EA000C73	B =0x000031F0
• 00000020: 00000000	ANDEQ r0,r0,r0
① 「Start」ク 000000 000 000000 0000 00000 00000	リック ANC ANC ANC ANC ANC ANC ANC ANC ANC ANC
00000 0000000 0000000	ANDEQ 0, r0, r0 ANDEO r0, r0, r0
• SB1 0000	▼ CLR ▼ 実行回数 1 2
G Brook Ret R4E	Ret N3 Win Roy Watch Sym Trace Step CIrac OStep Lints
u break instruer	
Start	Info Log Esc Stop

4-3-2. I-jet を使用(詳細なデバッグ操作は IAR 提供「デバッグガイド」を参照)

1) サンプル「Sample_BARE_M3」をダウンロードする。



2) デバッガ起動画面



4-4. デバッグに関する特記事項

- I-jet や DEFnano を使用してのデバッグに関する共通事項は、デバッグ時はプログラムコードを RZ/T1の内蔵 RAM にダウンロードしてデバッグ作業を進めます。
 特に JTAG デバッガを使用したサンプル例ですとダウンロード時にシリアル FROM に書き込み後、 アプリ側でロードさせるコードが含まれていますが、MP-RZT1-01を使用する場合は不要です。
- 2) MP-RZT1-01 基板は、オンボードのシリアル FROM にローダーとデバッガ用ファームが書き込み済 みで出荷しています。
- 3) ROM 書き込みに関しては、5項をご覧下さい。
5. ROM 書き込み

5-1. DEFnano を使用して ROM 書き込みする。

【オプション】-	【フラッシュ ROM	【ライタ】	を選択
----------	------------	-------	-----

存止 書記 080000	EI PC	C 0×2008000	0 [View C	PURZ/A		環境設定 CPU設定	(E) (C)			00800	100 SIZ 0×	200C7CE	i TargetMe	smory 🖕
20080000:	35	LOR ;	ec, ≕	Reset_ha	ndler		-	75.02		A DIE	~					
20080004:	36	LOR	PC, =	Undef ine	d_handle	er.	5	1000	TROM	71 2(1)	>	-				
20080008:	3/	LUK	PC, =	Svc_hand	ler bendler			ファーム	バージョ	ンアッフ	(V)	100				
20080010	39	LOR		Abort ba	ndler	51	-	* Star	+0~0010		ッフボート					
20080014	40	LOR		Reserved	bandle	- 12		Star	+0-0014	Rese	wed					
20080018:	41	LDR		Irg hand	ler			: Star	+0×0018	: 180						
2008001C:	42	LDR I	. =	Fig hand	ler			: Star	+0x001C	: FIQ()	MI)					
20080020:	43	;=======														
	44	; SFROM	言愛	録したロ	-5-15	渡す情報										
-	45	;*******														
	46	Info_tab	le													
-	47	DCD	In	age\$\$VEC	TOR_TABL	E\$\$Base		; Star	+0x0020	: 内蔽	All転送先	の開始アト	ドレス			
-	48	DCD	In	age\$\$DAT	A\$\$Limit	1		; Star	+0x0024	: 内蔽	AII転送先	の終了アト	レス(+1))		
	49	DCD	vec	tor_tabl	e			; Star	t+0x0028	: 初期	C值					
	50	Info_end														
	51	;*******														
	52	Literals														
	53	LTOR	3													
	54															
	55	END														
SB1 0000000	00			CLR] C 38	2 000000	0000		1	- CLR	• 実行	一個数 1		y	フトプレーク	BRK co
-	3				(なし)	· ch	iar	*	161	*	周期				
																_
Go Bres	k Ret	Mon	1	Win	Ree	Wate	h	Svn I	Irace	Sten	CIrer	CSten		IntE	Is Pro	E
00 0160	ix hot	non		0.111	105	I acc	0 0	5 Y W.	i ace	OLCH	VITA	outer		1000		9.C

(W+UsrAp¥C_H28_AICHI¥RZT1¥Sample_IAR¥RZT1_Sample_BARE¥M	3¥Debugi(参照)	
Start 0x0 end 0x977 Size 0x978 -Cortex-R4F(*.mot/*hex) ("N¥UsrAp¥C_H28_AICHI¥RZT1¥Sample_IAR¥RZT1_Demo_BARE¥R4F	¥Debug¥ (梦麗)	の Hex ファイ ルを指定す る。
Start 0x0 end 0x38aff Size 0x38b00 書込み個数 0 全消去個数 0	書込み開始 全消去開始	先頭を☑後、 Cortex-R4F側 の Hex ファイ











シリアルフラッシュ ROM マップ(MP-RZT1-01)					
ローダ・デバッグ用ファーム	0~1セクター				
	0x0~0x1_FFFF				
アプリケーションエリア	2~15セクター				
Cortex-M3	0x2_0000~0x9_FFFF				
	0xA_000~0xF_FFFF(予備)				
アプリケーションエリア	16~31セクター				
Cortex-R4F	0x10_0000~0x17_FFFF				
	0x18_0000~0x1F_FFFF(予備)				

- 6. 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および IAR 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。
- 7. 商標
 - ・EWARMは、IAR社の登録商標、または商品名称です。
 - ARM Cortex, Thumb および ARM Cortex-M3/R4F は ARM Limited の EU およびその他の国における商標および登録商標です
 - ・RZT1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。
- 8. 参考文献
 - ・「RZ/T1 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
 - ・「IDEプロジェクト管理およびビルドガイド」 IAR社
 - ・「IAR C/C++開発ガイドコンパイラおよびリンク」 IAR 社
 - ・「IARアセンブラリファレンスガイド」 IAR社
 - ・「IARデバッグプローブガイド」 IAR社
 - ・その他

$\mp 486-0852$

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