

RL78 ファミリ CS+ ホットプラグイン機能を用いたデバッグ方法

R20AN0248JJ0400 Rev.4.00 2023.12.20

要旨

本書は、RL78/Fxx(RL78/F12除く)が持つホットプラグイン機能を用いたデバッグ方法を説明するものです。

CS+ for CA, CX、Renesas Flash Programmer、E1 エミュレータ、ホットプラグアダプタ(ROE000010ACB00)、 CPU ボード(QB-R5F10BMG-TB)を使用し、プログラム作成→ビルド→デバッグの順にツール設定方法や接続 手順等を具体的に説明します。

CS+ for CC をご使用の場合も、ホットプラグインの接続方法は同じなので、本書の手順に沿ってご使用ください。ただし、<u>ホットプラグイン初期化関数</u>のみ異なるのでご注意ください。

備考 1: CS+の操作方法は、CS+のユーザーズマニュアルをご確認ください。

- 2: E1 エミュレータとユーザシステムとの接続回路や使用するユーザ資源の情報については、エミュ レータ ユーザーズマニュアル 別冊(文書番号: R20UT1994)をご確認ください。
- 3: Renesas Flash Programmer の操作方法は、Renesas Flash Programmer のユーザーズマニュアルを ご確認ください。



図 1. システムの概要



目次

1.	ホットプラグインを使用したデバッグの概要	3
1.1	ホットプラグインの特徴	3
1.2	ホットプラグイン接続後のデバッグ	4
2.	ホットプラグイン機能を用いたデバッグ方法	5
2.1	作業手順の概要	5
2.2	実践	6
2.2.1	ホットプラグインが可能なプログラムの作成	6
2.2.2	ビルドの実行	
2.2.3	プログラムの書き込みと実行	
2.2.4	ホットプラグイン接続	
2.2.5	デバッグ	
2.2.6	ユーザシステムとの切断	
3.	注意事項	
3.1	デバッグDTC動作クロックに関する注意事項	
3.2	DTCの保留命令に関する注意事項	
3.3	データアクセスイベント関する注意事項	
3.4	32bit長以上の変数アクセスに関する注意事項	
3.5	スタンバイモードに関する注意事項	
3.6	リセットに関する注意事項	
3.7	RAMの使用に関する注意事項	

1. ホットプラグインを使用したデバッグの概要

1.1 ホットプラグインの特徴

ホットプラグインとは、ユーザシステム動作中に"プログラム動作を継続したまま""リセットをかける ことなく""プログラムの内容を変更せずに"デバッガを接続できる機能です。

ユーザシステムの検査時や出荷後に不良が発生した際、ホットプラグインにより不良発生状態を保持した ままデバッグすることができます。

主な特徴は以下のとおりです。

●特徴1 不良発生状態を保持

- ・プログラムの動作を継続したまま接続可能
- ・リセットを発生させずに接続可能
- ・プログラムの内容を変更せずに接続可能
- ●特徴2 プログラムをセキュリティ機能で保護
 - ・デバッガ接続時に ID コード認証

●特徴3動作を継続したまま解析

・デバッガ接続後、CPU を止めずに DTC 方式の RAM モニタが可能

●特徴4 更なる詳細解析

- ・強制ブレークが使用可能
- ・ブレーク後、ソフトウェアブレークやイベントブレークの設定が可能
- ※ホットプラグインは、不良解析を目的とした機能なので、ユーザプログラム開発の初期段階では通常のオ ンチップデバッグ機能を使用してください。

1.2 ホットプラグイン接続後のデバッグ

ホットプラグインで動作中のマイコンヘデバッガを接続した後は、デバッグ専用の DTC を使用した RRM や DMM を使用して CPU を動作させたままデバッグを行うことができます。

従来の RL78 マイコンではデバッグ時のメモリアクセスに CPU を使用していましたが、デバッグ DTC の搭載 により CPU を使用せずにデバッグ時のメモリアクセスが可能となりました。

DTC 方式 RRM/DMM でアクセス可能な領域は以下の通りです。

[読み出し/書き込み可能な資源]

- RAM
- SFR

[読み出しのみ可能な資源]

- ・データフラッシュ(読み出し許可時のみ)
- ・ミラー領域
- ・汎用レジスタ



図 2. デバッグ DTC の概念図

[用語の説明]

・DTC:データトランスファコントローラ。CPUを使わずにメモリとメモリの間でデータを転送する機能

- ・RRM:リアルタイム RAM モニタ。CPU を停止せずに RAM/SFR の読み出しを行う機能
- ・DMM:ダイナミックメモリモディフィケーション。CPU を停止せずに RAM/SFR へ書き込みを行う機能

- ホットプラグイン機能を用いたデバッグ方法
 ホットプラグイン機能を用いたデバッグ方法を説明します。
- 2.1 作業手順の概要

以下の手順で作業を行います。

CS+をインストールしてから作業を行ってください。



2.2 実践

2.2.1 ホットプラグインが可能なプログラムの作成

(1) CS+の起動

[スタート]→[すべてのプログラム]→[Renesas Electronics CS+]→[CS+ for CA, CX]を選択して CS+を起 動します。

- (2) プロジェクトの作成
 - ・スタートパネルより新規プロジェクトを作成します。

3	新しいプロジェクトを作成する 新たにプロジェクトを作成します。 既存のプロジェクトに登録されているファイル構成を流用して、作成することも可能です。
①作成開始	「新しいプロジェクトを作成する」の[GO]ボタンを押下してください

・プロジェクトの設定を行います。

プロジェクト作成		×							
マイクロコントローラ(工):	RL78	•							
使用するマイクロコントローラ(<u>u</u>);								
二日の「「「「「「」」」」、「「」」、「「」」、「「」」、「「」、「」、「」、「」、「									
RL78/F14 (ROM:48) RL78/F14 (ROM:64) RL78/F14 (ROM:64) RL78/F14 (ROM:96) R5F10PGF(48pin	B) - 品種名 R5F 10PMF (内部 R0Mサイズ(以)イト196 (内部 RAMサイズ(以)イト18192)								
R5F10PMF(80pi									
プロジェクトの種類(<u>K</u>):	プロジェクトの種類(M) アプリケーション(CA78K0R)								
ブロジェクト:名(<u>N</u>):	sample								
作成場所(1):	CN	▼ 参照(<u>R</u>)							
	V 709101-207+140								
C:¥sample¥sample.mtpj									
既存のブロジェクトのファイル構成を流用する(S)									
流用元のプロジェクト(P): (流用元のプロジェクト・ファイルを入力・マイチャン) ・ 参照(W)									
□ フロジェクト・フォルダ以下の構成ファイルをコピーして流用する(2) (4)									
	1 2 2 2 2 7 7 2 7 2 7 2 7 2 7 2 7 2 7 2								

① [使用するマイクロコントローラ]の指 定	使用するデバイス名を選択します。ここでは 「R5F10PMF(80pin)」を選択しています。
② [プロジェクト名]の指定	任意の指定が可能です。ここでは「sample」を 指定しています。
 ③ [作成場所]の指定 	任意の指定が可能です。ここでは「C:¥」を指 定しています。
④プロジェクトの生成	[作成]ボタンを押下してください。

- (3) ユーザプログラムの作成
 - ・main.cファイルを作成します。



①「ファイル」のコンテキストメ	プロジェクトツリー内の「ファイル」を選択後、右クリッ
ニューを開く	クしてください。
②新規ファイルを追加	「追加」→「新しいファイルを追加」を選択してください。
④ [ファイル名]の指定	任意の指定が可能です。ここでは「main.c」を指定してい ます。
⑤ [作成場所]の指定	任意の指定が可能です。ここでは「C:¥sample¥」を指定し ています。
⑤ファイル生成	[OK]ボタンを押下してください。

・main.cファイルをエディットします。



nain.cファイルにエディットしてください。
<u> テコビー&ベーストしてください。)</u>

- (4) ホットプラグイン用初期設定関数の作成
 - ホットプラグイン用初期化関数(hpi_initialize_prog) 定義ファイルを作成します。
 プロジェクトに hpi_init. asm ファイルを追加してください。

プロジェクト・ツリー	🕂 🗶 🚰 プロパティ 📝 main.c*		フライル追加	
ê 🕜 🙎 🗷	11 1 1 - つい カラム	•	ファイリン旦加	
 ■ 「読 sample (プロジュ ■ R5F10PMF (マ CA78KOR (ビ) RL78 シミュレ プログラム線 	<u>エクト)*</u> 行 マイクロコ 1 int g_1=0: ルド・ツー 3 int g_k=0: ノータ (デ) 4 5 Elvoid main 0 {	2	ファイルの理想は(」): 「シーン・ファイル(*c) ヘッジ・ファイル(*c) アゼンブリ・ファイル(*sam) リンク・ディレラティブ・ファイル(*ar;*d 変乱/開設が資料()/(*sy) Pythonスクリプト・ファイル(*sy)	ir)
追加) (<u>D</u>)	〕 既存のファイルを追加(E)…	J+AP JP1 M*IX0	
- <mark>9</mark> п 🖭 💷 та	クスプローラでフォルダを開く(<u>E</u>)	新しいファイルを追加(N)		
🔳 Wir	indowsエクスプローラのメニュー(<u>W</u>)	新しいカテゴリを追加(C)		<u>a</u>
	ロジェクトから外す(<u>R</u>) Shift+Delete		空のアセンブル・ファイルです。	
1 = E	ピー(<u>C</u>) Ctrl+C		ファイル名(<u>N</u>): pi_init.asm	
(語) 貼り	り付け(<u>P</u>) Ctrl+V		作成場所(<u>L</u>): >¥sample¥ -	
· 名前	前の変更(<u>N</u>) F2			
🔂 プ C	ロパティ(<u>P</u>)			K キャンセル ヘルブ(出)
		I	()	

①「ファイル」のコンテキストメ	プロジェクトツリー内の「ファイル」を選択後、右クリッ
ニューを開く	クしてください。
②新規ファイルを追加	「追加」→「新しいファイルを追加」を選択してください。
③[ファイル名]の指定	任意の指定が可能です。ここでは「hpi_init.asm」を指定 しています。
④[作成場所]の指定	任意の指定が可能です。ここでは「C:¥sample¥」を指定し ています。
⑤ファイル生成	[OK]ボタンを押下してください。

・ホットプラグイン用初期化関数(hpi_initialize_prog)をエディットします。



※下記内容は、ホットプラグイン初期化関数です。デバッグ専用 DTC の初期化/有効化を行っています。 そのままユーザプログラムヘコピーしてください。

[初期化関数の使用方法]

- ・RAM の初期化完了後、なるべく早いタイミングで呼び出してください。 CPU リセット解除後 400ms 内に初期化関数を実行完了してください。実行完了しない場合、ホット プラグイン後の端子リセット/POC リセット発生でデバッガがハングアップします。
- ・リセット後1回のみの呼び出しとしてください。
- ・初期化内で SFR を操作している箇所がありますので、SFR ガード機能を OFF にしてから呼び出して ください。
- ・掲載の初期化関数では、関数実行完了まで割り込みを禁止しています。割り込み禁止としたくない 場合は"for disable interrupt"のコメントがある3行を削除してください。(割り込み許可とす る場合、初期化関数実行完了までの時間やスタック使用量にご注意ください。)

	CS+	for	CA. CX	((RL78/F1x)	の場合
--	-----	-----	--------	-------------	-----

PUBLIC	_hpi_init	ialize_prog					
hni init	tialize pr	Og :					
	PUSH	PSW	for disable	interrunt			
	DI	101	for disable	interrupt			
	PUSH	AX		meenape			
	PUSH	DF					
	PUSH	H					
	MOV	A FS					
	PUSH	AX					
	MOV	ES. #000H					
	MOVW	DE, #000C4	Η				
	MOVW	HL, #0072	2H				
	BT	[HL].7,\$S	ET_TRCSIZE1				
SET_TRCS	IZE0:						
-	MOVW	HL, #00D00	H				
	BR	\$SEC_CODE	_SET				
SET_TRCS	IZE1:						
	MOVW	HL, #00E00	Н				
SEC_CHK:							
	MOV	A, !0723H					
	BT	A. O, \$DTC	_DES_SET				
SEC_CODE_	_SET :						
	MOV	A, ES: [DE]					
	MOV	[HL], A					
	INCW	DE					
	INCW	HL					
	MOVW	AX, DE					
	CMPW	AX, #000CE	H				
	SKZ						
	BR	\$SEC_CODE	_SET				
DTC_DES_S	SET:						
	MOV	ES, #00FH					
	MOV	L, #00EH					
	MOVW	AX, #OFFAD	Н				
	MOVW	[HL], AX					
	MOV	L, #010H					
DIC_SEL:			#0101				
	MOV	[HL+UUH],	#UI3H #001U				
	MOV	[IIL+VIH],	#0010				
	MOV	[IIL+U2H],					
			#000H				
	MOV	[NL+04H],	#000H				
	MOV	А, П [Ш.±ОБЦ]	•				
	MOV		H022H				
	MOV	[ΠL+V0Π],	#0230 #0074				
	MOV	[IIL+V/II],	#00/II				
	WUV	[⊓L+∪ŏH],	#0000				



MOV	[HL+09H], #001H
MOV	[HL+0AH], #001H
MOV	[HL+0BH], #001H
MOV	[HL+0CH], #023H
MOV	[HL+0DH], #007H
MOV	[HL+OEH], #OADH
MOV	[HL+OFH], #OFFH
MOV	[HL+10H], #011H
MOV	[HL+11H], #001H
MOV	[HL+12H], #010H
MOV	[HL+13H], #010H
MOV	[HL+14H], #OADH
MOV	[HL+15H], #OFFH
MOV	[HL+16H], #000H
MOV	A, H
MOV	[HL+17H], A
MOV	[HL+18H], #003H
MOV	[HL+19H], #001H
MOV	[HL+1AH], #010H
MOV	[HL+1BH], #010H
MOV	[HL+1CH], #000H
MOV	A, H
MOV	[HL+1DH], A
MOV	[HL+1EH], #OADH
MOV	[HL+1FH], #OFFH
SET1	DTCEN
POP	AX
MOV	ES, A
POP	HL
POP	DE
POP	AX
POP	PSW ;for disable interrupt
RET	
END	

CS+ for CC (RL78/F1x) の場合

.PUBLIC _hpi_i	nitialize_prog	g					
_hpi_initializ PUSH DI	e_prog: † PSW	;for ;for	disable disable	interrupt interrupt			
PUSH	I AX I DE	,		ape			
PUSH HL							
MOV A, ES							
PUSH	1 AX FS #0x000						
MOVW	DE, #0x000C4						
MOVW	HL, #0x00722	001751					
BT L SET	TRCSIZEO:	6917E1					
MOVW HL, #0x00D00							
BR \$	SEC_CODE_SET						
MOVW	V HL, #0x00E00						
SEC_CHK:							
MOV BT A	A, 10x0723	0 E T					
SEC_CODE_SET:	1.0, UTO_DLO_	JLI					
MOV	A, ES: [DE]						
MOV	[HL], A V DE						
INCW	V HL						
MOVW	I AX, DE						
CMPW SK7	I AX, #0x000CE						
BR \$	SEC_CODE_SET						
DTC_DES_SET:							
MOV MOV	ES, #0x00F						
MOVW	AX, #0x0FFAD						
MOVW	/ [HL], AX						

	MOV L #0x010	
DTO SET.		
DIC_SEL	MOV [111.000] #0010	
	MUV [HL+UXUU], #UXUI3	
	MOV [HL+0x01], #0x001	
	MOV [HL+0x02], #0x00A	
	MOV [HL+0x03], #0x00A	
	MOV [HL+0x04], #0x000	
	MOV A, H	
	MOV [HL+0x05], A	
	MOV [HL+0x06], #0x023	
	MOV [HI +0x07] #0x007	
	MOV [HI +0x08] #0x000	
	MOV [HL+0×00] #0×000	
	MOV [HL+0x03], #0x001	
	MUV [HL+UXUC], #UXU23	
	MOV [HL+0x0D], #0x00/	
	MOV [HL+0x0E], #0x0AD	
	MOV [HL+0xOF], #0xOFF	
	MOV [HL+0x10], #0x011	
	MOV [HL+0x11], #0x001	
	MOV [HL+0x12], #0x010	
	MOV [HL+0x13], #0x010	
	MOV [HL+0x14], #0x0AD	
	MOV [HL+0x15], #0x0FF	
	MOV [HL+0x16], #0x000	
	MOVAH	
	MOV [HI +0x17] A	
	MOV [HL+0x18] #0x003	
	MOV [HL+0v10], #0v000	
	MOV [IIL:0x13], #0x001	
	MOV [HL+UXIA], #UXUIU	
	MOV [HL+UXIB], #UXUIC	
	MOV [HL+UXIC], #UXUUC	
	MOV A, H	
	MOV [HL+0x1D], A	
	MOV [HL+0x1E], #0x0AD	
	MOV [HL+0x1F], #0x0FF	
	SET1 !DTCEN	
	POP AX	
	MOV ES, A	
	POP HL	
	POP DE	
	POP AX	
	POP PSW	for disable interrunt
	RFT	

CS+ for CC (RL78/F2x) の場合

.PUBLIC _hpi_initialize_prog_f2x				
hpi initialize prog f2x:				
PUSH PSW	;for	disable	interrupt	
DI	;for	disable	interrupt	
PUSH AX				
PUSH DE				
PUSH HL				
MOV A, ES				
PUSH AX				
MOV ES, #0x000				
MOVW DE, #0x000C4				
MOV A, ES: [DE]				
BT A. 2, \$GO_SETHPI				
BR \$!NOHPISET_RETURN				
GO_SETHPI:				
MOVW DE, #0x000C6				
MOVW HL, #0x00722				
BT [HL]. 7, \$SET_TRCSI	ZE1			
SET_TRCSIZE0:				
MOVW HL, #0x00D00				
BR \$SEC_CHK				
SET_TRCSIZE1:				
MOVW HL, #0x00E00				
SEC_CHK:				
MOV A, !0x0723				

BT A.O, \$DTC_DES_SET
SEC_CODE_SET:
MOV A, ES. [DE] MOV [HI] A
INCW DE
INCW HL
MOVW AX, DE
CMPW AX, #0x000D6
SKZ
BR \$SEC_CODE_SET
DTO DES SET
MOV ES #0x00F
MOV L, #0x00E
MOVW AX, #0x0FFAD
MOVW [HL], AX
DTC_SET_SUB:
MUV L, #UXUIU
MOV [HI +0x00] #0x013
MOV [HL+0x00], #0x010
MOV [HL+0x02], #0x010
MOV [HL+0x03], #0x010
MOV [HL+0x04], #0x000
MOV A, H
MOV [HL+0x06] #0x023
MOV [HL+0x07], #0x020
MOV [HL+0x08], #0x000
MOV [HL+0x09], #0x001
MOV [HL+0x0A], #0x001
MOV [HL+0x0B], #0x001
MOV [HL+0x00], #0x023 MOV [HL+0x0D] #0x007
MOV [HL+0x0E], #0x0AD
MOV [HL+0x0F], #0x0FF
MOV [HL+0x10], #0x011
MOV [HL+0x11], #0x001
MOV [HL+0x12], #0x010 MOV [HL+0x13] #0x010
MOV [HL+0x14], #0x040
MOV [HL+0x15], #0x0FF
MOV [HL+0x16], #0x000
MOV A, H
MUV [HL+UXI/],A MOV [HL+Ox19] #0x002
MOV [HL+0x19], #0x003
MOV [HL+0x1A], #0x010
MOV [HL+0x1B], #0x010
MOV [HL+0x1C], #0x000
MOV A, H
MOV [HL+OXID], A MOV [HL+Ox1E] #OxOAD
MOV [HI +0x1E], #0x0AD
SET1 IDTCEN
NOHPISET_RETURN:
POP AX
MOV ES, A
POP DE
POP AX
POP PSW ;for disable interrupt
RET

プロジェクト・ツリー	🗜 🗙 🚰 ታዐ/የምሩ 🍯 main.c 📑 hpi_init.asm
2 🕜 🙎 🛛 🗃	111 動 ⇒ つ い カラム・
Gample (プロジェクト)* Rr78 シミュレータ (デ) Si スタートアップ Gamma C main.c main.c	if image: space spac
①「main.c」を開く	プロジェクトツリー内の「main.c」をダブルクリックし
	てください。
②ホットプラグイン初期化関数を 記述	下記内容をmain.cファイル内に追加してください。
void hpi_initialize_prog(void);	
int g_i=0; int g_j=0; int g_k=0;	今回の追加内容
void main() {	
<pre>hpi_initialize_prog(); while(1) { g_i++; g_j++; g_k++; }</pre>	今回の追加内容

・main 関数内でホットプラグイン用初期化関数(hpi_initialize_prog)を呼び出す記述をします。



2.2.2 ビルドの実行

- (1) セキュリティ ID の設定
 - ・セキュリティ ID の設定を行います。

ビルドツールのプロパティにある共通オプションで設定します。

	4 ×	🚰 プロパティ 📝 main.	.c* Mpi init.asm*
9 🕜 🧏 😰		へ CA78K0R のプロパラ	4
Comple (プロジェクト)		□ Fuk·+-k	
	-	ビルド・モード	
- RSF10PMF (74.20		□ 出力ファイルの種類とす	易所
CA/8KOR (EJLF · y	E PILK	■ <u> 山田コーノルの強考</u> ・ プロミニケクト(B)	E7
🛫 プログラム解析 (解析)	😽 リビル	,ド・プロジェクト(<u>R</u>)	Shift+F7
⋴-勐 ファイル	🔜 クリー	ン・プロジェクト(C)	
💕 スタートアップ		> >=>=>+(<u>=</u>)	
	🎦 現在の	ビルド・オプションを	sample の標準に設定する(<u>S</u>)
hpi_init.asm*	TT PILK	・オプションの 🕥	- b(I)
	A		1 (±/
	🖳 リンク	順を設定する()	
		₹	
		2 1 (<u>E</u>)	
		出力フォルダ	
		∎ ш+h ¬=,/ il /⁄	
🔨 CA78K0R のプロパティ			
 CA78K0R のプロパティ 定義マクロ 			定義マクロ[0]
 CA78K0Rのプロパティ 定義マクロ よ(使うオジョン(アセンブラ) 			定義マクロ[0]
 CA78K0R のプロパティ 定義マクロ よ(使)オブション(アセンブラ) 追加のインクルード・パス 			定義マクロ[0] 追加のインクルード・パス [0]
 CA78K0Rのプロパティ 定義マクロ よ(使)オブション(アセップラ) 追加のインクルード・パス システム・インクルード・パス マニネックロ 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] テ美マクロ [0]
 CA78K0Rのプロパティ 定義マクロ 太(使カブション(アセンブラ) 道加のインクルード・パス システム・インクルード・パス 定義マクロ メ(使サオブション(印) 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0]
 CA78K0Rのプロパティ 定義マクロ よ(使オブション(アセンブラ) 追加のインクルード・パス システム・インクルード・パス 定義マクロ よ(使)オブション(の) 使用するライブラリ・ファイル 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0]
 ▲ CA78K0Rのプロパティ > 定義や20 > はかみインクルード・パス > システム・インクルード・パス > システム・インクルード・パス > 定義や70 > よ(使)オブシシン(の) > (使用するライブジ)・アイル > (透明のライブジ)・アイス 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・パス[0]
 ▲ CA78KORのプロパティ > 定義マクロ > よ(な)オ/ション(アセップラ) > 追加のインクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > 定義マクロ > よ(せ)オプション(の) > (使用するライブラリ・アイル > 追加のライブラリ・パス 出力フォルダ 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] (使用するライブラリ・アマイル[0] 追加のライブラリ・パス[0] KBuildModeName&
 ▲ CA78KORのプロパティ > 定義マクロ > は使うオブション(アセンプラ) > 道面のインクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > 送気マクロ > はまた、システム・インクルード・パス > 使用するライブラ・アイル > 道面のうイブラレ・パス 出力フォルダ 出力フォールダ 出力フォールダ 出力フォールダ 			定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] NBuildModeNameS/ MProjectMameS/Imf
 ▲ CA78KORのプロパティ > 定義や20 > はかのインクルード・パス > システム・イングルード・パス > システム・イングルード・パス > 定義や70 > メ化センオブシュン(ソング) > 使用するライブジリ・フィイル > は加のライブジリ・フィール > はのうイブジリ・アイル名 > メイセンオブシュン(ROM(ヒプロ) > メイセンオブシュン(ROM(ヒプロ) 	(29 ^t)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] 終起uildModeName% %ProjectName%Imf
 ▲ CA79KOR のプロパティ > 定義マクロ > よ(センオブシュン(アセンブラ) > 追加のインクルード・パス > システム・イングルード・パス > ご案等クロ > よ(センオブラン(アセンブラン) > 使用するライブラいパス 出力フォルボ 出力ファイル名 > よくビングラン(ROMEプロ ROM(ビ用オブラン)・ファイル > レビステンス(ROMEプロ ROM(ビ用オブラン)・ファイル 	セッサ) ・を出力する - マンザーク)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・アマイル[0] 追加のプイブラリ・パス[0] XBuildModeName# NProjectName#Limf いいえ
 ▲ CA78K0Rのプロパティ > 定義マクロ > は使うオブション(アセンブラ) > 道筋ののインクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > ジステム・クンクルード・パス > 逆義でするシイブションマイル > 値面のライブラリ・パス > 出力フォルボ 出力ファイルを > ス(使)オプション(ACMLプロ ROM(L用すびシュント・ファイル > 本(使)オプション(スイブジョン) > 本とサ・ファイルを > ス(使)オプション(スイブジョン) 	セッサ) レを出力する - エバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] メBruidMode Name& NProjectName&Imt (パいえ (パい、
 ▲ CA79K0Rのプロパティ > 定義マクロ > よ(た)オブション(アセンブラ) > 追加のインクルード・パス > システム・イングルード・パス > テ定義マクロ > よ(た)オブション(アイル) > 使用するライブラリ・アイル > 追加のライブラリ・パス > 出力フォルダ > 出力フォルダ > 出力フォルダ > よくた)オブション(オブラュン) へ おサ・ファイル 思力する ヘキサ・ファイルを出力する。 	セッサ) いを出力する ・ エバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] メロルIdModeName& メProjectName&Limf いいえ はい メBuildModeName&
 ▲ CA78K0Rのクロパティ > 定義マクロ > は使うオブション(アセンブラ) > 適加のインクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > 逆義 マクロ > よ(使うオブション(ソカ) > 使用 するライブラリ・ファイル > は知つうイブラリ・ノス > 出力ファイル名 > よ(使うオブション(スペロガンロ) > よ(使うオブション(スペロガンロ) > よ(使うオブション(スペロガンロ) > よ(使うオブション(スペロガンロ) > 大(サンディション) > ペキサ・ファイル名 > ペキサ・ファイル名 	セッサ) いを出力する ・ エバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のプイブラリ・パス [0] XBuildMode NameX NProjectNameKimf (べいえ はい XBuildMode NameX NProjectNameKimg
 ▲ CA78K0Rのプロパティ > 定義マクロ > は使うオブション(アセンブラ) > ibmのインクルード・パス > システム・インクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > 送表・イブラン・ファイル > ibmのうイブラン・パス > 出力ファイル名 × は使うオブション(スログシロ) > ス体プンブンイル23 > ペキサ・ファイル・出力フォルダ ヘキサ・ファイル・ピューマット 	セッサ) を出力する ・ ンバータ		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] としていたいのである WProjectNameXimm (くいえ (ない XBuildModeNameX WProjectNameXimm (くいえ (ない XBuildModeNameX WProjectNameXimm (くいえ
 ▲ CA79KORのプロパティ > 定義マクロ > よ(た)オブション(アセンブラ) > 追加のインクルード・パス > システム・インクルード・パス > テステム・マンクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > 逆第のライブラリ・パス > 送加のライブラリ・パス > 出力フォルダ > 出力フォルダ > 出力フォルダ > 出力フォルダ > 出力フォルダ > 出力フォルダ > ス(た)オブション(オブラェン) へ 本サ・ファイル と、 > へ キサ・ファイル と、 > ヘ キサ・ファイル ション ホーメーシャーメーシャート > マ デバイス 	セッサ) - た出力する - ・ エンバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] 地域のライブラリ・ファイル[0] KBuildModeName& MProjectName&Imf いいえ はい KBuildModeName& MProjectName&Nex MProjectName&Name&Name&Name MProjectName&Name&Name
 ▲ CA78K0Rのクロパティ > 定義マクロ > は使うオブション(アセンブラ) > 適加のインクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > ジステム・インクルード・パス > 逆義マクロ > よく使うオブション(ワイム) > 使用するライブラリ・ファイル > は回知のイブラリ・パス > 出力ファイル名 > よく使うオブション(ストロー) > ホキサ・ファイル・フォーマット > デドイス > ビキュン(アー) 	セッサ) を出力する - エンバータ		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・アテイル[0] 追加のライブラリ・アテイル[0] はしいのといいののののののののののののののののののののののののののののののののの
 ▲ CA78K0Rのプロパティ > 定義マクロ > は使うオブション(アセンブラ) > 道動のインクルード・パス > システム・インクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > ジスピングランへのいたい > 使用するシイグリ・ファイル > は、使うオブシント・ファイル > メ、使うオブシント・ファイル > メ、使うオブシント・ファイル > メ、使うオブシント・ファイル > メ、使うオブシント・ファイル > メ、キサ・ファイル・ジューマット > デバイス > ビルド方法 	セッサ) を出力する - エバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] 追加のライブラリ・パス[0] KBuildModeNameX KProjectNameXimt (い、え はい XBuildModeNameX KProjectNameXimt (い、え して、)
 ▲ CA79KUR のプロパティ > 定義マクロ > よ(使)オブション(アビップラ) > 追加のインクルード・パス > システム・インクルード・パス > ジステム・インクルード・パス > ジステンクルード・パス > ジステンクルード・パス > ジステンクルード・パス > ジステンクルード・パス > ジステンクルード・フィール > ジステンティール・フォーマット > デバイス > ビルド方法 > メスト・フィンク 	セッサ) を出力する - <u></u><u></u>レパータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] 地域のタイズラリ・ファイル[0] 米DuildModeName& メProjectName&Imf いいえ はい 大型uildModeName& メProjectName&Nex いて、 にくたie 1000000000000000000000000000000000000
	セッサ) を出力する - エンバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] メロロロのライブラリ・ファイル[0] にない。 メProjectName%Imf にない。 ない。 ない。 ない。 ない。 ない。 ない。 ない。
 ▲ CA78KOR のプロパティ > 定義マクロ > は使うオブション(アセンブラ) > ibmのインクルード・パス > システム・インクルード・パス > はた、オンクシード・パス > はた、オンクシーンマイル > はかつライブション(スポンク・アイル > はた、オンシント・ファイル > はた、オンシント・ファイル > はた、オン・ファイル・出力・ファイル・出力・オージ・ファイル・出力・オージ・ファイル・シュージー・シュージ > ビルド方法 > こみ > こみ ○ (シーシン) 	セッサ) 、を出力する ・ コンバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ウァイル[0] 追加のライブラリ・ウァイル[0] 追加のライブラリ・パス(0) KBuildModeNameX KProjectNameXimeX (い、え はい KBuildModeNameX KProjectNameXimeX (い、え して、)
 ▲ CA78KOR のプロパティ > 定義マクロ > よ(使)オブション(アビップラ) > 追加のインクルード・パス > ジスティインクルード・パス > ジスティインクルード・パス > ジスティインクルード・パス > ジェ義マクロ > よ(使)オブション(アイル > 送加つライブラ)・パス 出フファイルを × よ(使)オブション(ACMEプロ ROM(ビ用オブジュント・ファイル > よ(使)オブション(オブジェント ヘキサ・ファイルを出力する) > よ(使)オブション(オブジェント ヘキサ・ファイルと出力する) > ベトボーンアイル > ボーンアイル > ビルド方法 > パージョン選択 ③ ○ ビルド・モード 	セッサ) 、を出力する ・ エバータ)		定義マクロ[0] 追加のインクルード・パス [0] システム・インクルード・パス [0] 定義マクロ [0] 使用するライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] 追加のライブラリ・ファイル[0] KBuildModeName& WProjectName&Imf いいえ はい 安臣の000000000000000000000000000000000000

① CA78KOR(ビルドツール)」のコンテキスト	プロジェクトツリー内の「CA78KOR(ビルドツール)」を
メニューを開く	選択後、右クリックしてください。
②プロパティを開く	コンテキストメニュー内の「プロパティ」を選択して
	ください。
③[共通オプション]タブを開く	メインパネルにある[共通オプション]タブを選択して
	ください。
④「セキュリティ ID」の設定	「デバイス」項目にある[セキュリティ ID]を指定しま
	す。任意の指定が可能です。ここでは
	「000000000000000000」を指定しています。

- (2) オプションバイトの設定
 - ・オンチップデバッグオプションバイトの設定を行います。
 ビルドツールのプロパティにあるリンクオプションで設定します。



①「CA78KOR(ビルドツール)」のコンテキスト	プロジェクトツリー内の「CA78KOR(ビルドツー
メニューを開く	ル)」を選択後、右クリックしてください。
②プロパティを開く	コンテキストメニュー内の「プロパティ」を選択し
	てください。
③[リンクオプション]タブを開く	メインパネルにある[リンクオプション]タブを選
	択してください。
④「デバイス」項目を開く	「デバイス」項目を開いてください。
⑤[オンチップデバッグの許可/禁止をリンクオプ	「はい」を選択してください。
ションで設定する〕	
⑥[オンチップデバッグオプションバイト制御値]	「87」を指定してください。
⑦[デバッグモニタ領域サイズ]	「0」を指定してください。
	※ホットプラグインでない通常のオンチップデ
	バッグを行う際はサイズを指定します。

[オンチップデバッグオプションバイト設定について]

ホットプラグインを使用する場合、オンチップデバッグオプションバイトには 0x87 を設定する必要があり ます。ホットプラグイン検出には低速オンチップオシレータを使用するため、この設定を行うと低速オン チップオシレータが動作し、ユーザプログラムによる停止ができなくなります。ただし、スタンバイモー ド中はレジスタ設定により低速オンチップオシレータを停止可能です。 ・ユーザオプションバイトの設定を行います。

ビルドツールのプロパティにあるリンクオプションで設定します。

4	ライブラリ /#用する= / ブラル フェ / ル	(市田オスニ / ゴニリ・コニ / リ [0]
	(大田 9 の ノイ ノ ノゲー ノゲー ノル	「使用するブイブブリ・ファイル[0]
Ľ.	Sebo A STANDA AND A SECONDARY	シスチム・ディテノティアドリルし 注意わからイゴニリ・パフ [0]
Ľ,	31777777777777777	317575454187[0]
		2X7X 71777 1X60
11	ノハイス オンチボッガの時間 /茶店大口、カナギバットで設定する	(1()(-ma)
	オンチップ・デバッグ・オブション・バイト制御値	14C1 607
	オンテップ・フィッジ・オフション・パイトが同時間	17500
	リアリック・モニング現代開始アドレス	
	フレザップにとう時間のフィスレイトロ	
	ユージ パンション パー ときべとう シー	HEX SOFFER
	2 9 99929 7 mm 12 ミラー領域指定	MAA=0(-mi0)
	フラッシュ・フタート・アドレスを設定する	いいえ
	ブート領域用ロード・チジュール・ファイルタ	0.032
	ヤルフRAM領域への配置を制御する	いいえ
	トレースRAM領域への配置を制御する	いいえ
	ホット・プラグインBAM領域への配置を制御する	いいえ
⊳	101-7	0.0.32
Þ	スタック	
בי בי	ーサ・オブション・バイトを設定する。 ーサ・オブション・バイトに値を設定する場合に指定します。	
-el	オフションに相当します。	

①[ユーザオプションバイトを設定する]	「はい」を選択してください。
②[ユーザオプションバイト値]	任意の指定が可能です。ここでは「60FFE8」を指 定しています。

- (3) ホットプラグイン用 RAM の確保
 - ・ホットプラグインで使用する RAM 空間をユーザプログラムが使用しないように
 ビルドツールのプロパティにあるリンクオプションで設定します。
 ※ホットプラグイン用 RAM 領域への RAM 確保が不必要な品種の場合、この項目は CS+にありません。設定不要ですので次の手順に進んでください。

CHYOKUK W/U/J/	
フ1フフリ (使用オステノゴー)・ファノリ	使用するこくづつしってくいい。
大田 9 9 71 7 77 771 //	反用9のブイノブゲー/2003
SATA 21 JUL 12	2001 201 201 201 201 201 201 201 201 201
2010/071 ノブリイベム	1日月10月7日 ノフリ・ハス[0]
2476-2172774	272721222217222777200
テノリス	$(t_{1})(-)$
オフナップ・ナハックの計画/茶店をリフク・オフンヨノに設定する	(JU) (-80)
オンナップ・ナハッジ・オノンヨン・ハイト市川岬世	HEN 8/
ティック・モニシャ県-取用的にアトレス	HER T/EUU
テハック・モーダ領域サイスレハイト」	
ユーザ・オノンヨン・ハイトを設定する	Lali(-gD)
ユーザ・オフション・ハイト1世	HEN DUFFE8
ミフト領域指定	MAA=U(-mi0)
フラッシュ・人ダート・アドレ人を設定する	00z (1)
フート領域用ロート・センュール・ファイル・冶	
セルフRAM領域への配置を制御する	nnz V
トレー人RAM関数への配置を制御する	
ホット・プラクインRAM領域への配置を制御する	はい(エラーを表示)(-ocdhp
メッセーシ	
スタック	
ット・ブラグインRAM領域への配置を制御する	
ット・プラヴインRAM領域への配置制御を指定します。"はい(エラーを表示)(-ocd	hpi)"指定時には、ホット・プラグインRAM領域へ
ホット・プラグインRAM領域への配置時にワーニングを出力します。"いいえ"指定	目時には、ホット・プラグインRAM領域を内部RAM
odhpi/-oodhpiwオブションに相当します。なお、本オブションを指定すると、-self.	/-selfw、-ocdtr/-ocdtrwも指定されたとみなされ

 ①[ホットプラグイン RAM 領域への 配慮を制御する]

域への 「はい(エラーを表示)」を選択してください。

(4) ビルドの実行

・エディットとビルドツールのオプション設定が終了したのでビルドを実行します。





2.2.3 プログラムの書き込みと実行

(1) プログラムの書き込み

・Renesas Flash Programmer (ライタ) にて、作成したヘキサファイルを対象デバイスに書き込みます。

Renesas Flash Programmer の使用方法は、[スタート]→[すべてのプログラム]→

[Renesas Electronics Utilities]→[書き込みツール] →[Renesas Flash Programmer Vx.xx]内にある ユーザーズマニュアルをご参照ください。



- (2) プログラムの実行
 - ・ユーザシステムに電源を供給し実機動作を行います。



2.2.4 ホットプラグイン接続

- (1) ホットプラグアダプタ(ROE000010ACB00)とユーザシステムの接続
 - ・最初に、必ず R0E000010ACB00 をユーザシステムに装着します。
 R0E000010ACB00 は、E1 エミュレータと接続する際に必ず GND から接続できるような機構となっています。



(2) E1 エミュレータとホスト PC との接続

・E1 エミュレータとホスト PC を接続します。



- (3) デバッグ機能の設定
 - ・CS+にて、デバッグツールを選択します。

プロジェクト・ツリー 🛛 🕂 🗙	🚰 プロパティ 📝 main.c 📝 hpi_	init.asm
2 🕜 🙎 🛢	11111 🖄 🔿 🔿 🔊 カラム	↓
Sample (プロジェクト)* ■ R5F10PMF (マイクロコントローラ CA78KOR (ピルド・ツール)	行 1 2 void hpi_initialize_p 3	r <mark>(②</mark> :
	するデバッグ・ツール(<u>D</u>) →	RL78 IECUBE(I)
□ ① ファイル	パティ(<u>P</u>)	RL78 E1(Serial)(<u>L</u>)
■ 🗊 ビルド・ツール生成ファイル	9	RL78 E20(Serial)(<u>R</u>)
	10 hpi_initia 11 while(1) {	RL78 EZ Emulator(Z)
🔤 hpi_init.asm	12 13 g	RL78 シミュレータ(<u>5</u>)
	14 g_k++ 15 }	;
	16	

①「デバッグツール」のコンテキ	プロジェクトツリー内の「デバッグツール」を選択後、
ストメニューを開く	右クリックしてください。
②デバッグツールの選択	「使用するデバッグツール」→「RL78 E1(Serial)」を選 択してください。

・E1 エミュレータ接続時の認証用セキュリティ ID を設定します。また、ホットプラグインのリトライ設定 を行います。



 デバッグツール」のプロパティ パネルを開く 	プロジェクトツリー内の「デバッグツール」を選択後、 右クリックしてください。
 認証用のセキュリティ ID の 入力 	[接続用設定]タブ内の「フラッシュ」項目にある[セキュ リティ ID]を指定します。 ビルドツールのオプションで設定した値と同じ値を入力 します。ここでは「00000000000000000000000000000000000
 ③ ホットプラグインの リトライ設定 	CPU が STOP モード中などで低速オンチップオシレータが 停止しているために、ホットプラグイン接続に失敗した 場合のリトライ間隔とリトライ回数を設定します。 今回のプログラムでは低速オンチップオシレータを停止 しないので、デフォルト設定(1000ms、3回)のままで使 用しています

[ホットプラグインのリトライ設定について]

・リトライ間隔にはプログラム内の STOP モード最大継続時間を入力してください。 ただし、リトライ処理中(リトライ間隔×リトライ回数)はデバッガが操作できない状態となります。 このため、リトライ処理の時間が極端に長くならないようにご注意ください。



・RAM モニタ間隔を設定します。



①「デバッグツール」のコンテキ	プロジェクトツリー内の「デバッグツール」を選択後、
ストメニューを開く	右クリックしてください。
②「プロパティ」を開く	「プロパティ」を選択してください。
③[表示更新間隔]の指定	[デバッグツール]タブ内の「実行中のメモリアクセス」
	項目にある[表示更新間隔]を指定します。
	100msc 以上の値が指定可能です。
	ここでは「500」を指定しています。

(4) ホットプラグイン接続の実行

・ホットプラグイン接続の準備を行います。

:クト(<u>P</u>) ビルド(B <mark> </mark> デノ	<u> 「ッグ(D)</u> ソール(<u>T</u>) ウインドウ(<u>W</u>) ヘルプ(<u>H</u>)	
ه 🗠 ۲ 🖓 🖞	デバック ツールヘダウンロード(<u>D</u>)	*
× 🔤 วือเพิรา 🐻	ビルド&デバック・ハヘダウンロード(<u>B</u>) F6	
🔊 RL78 E1(Seri 💼	デバッグ・ツール 接続(<u>C</u>)	
	ホット・プラグイン(<u>H</u>)	[0]
-= メモリ書き込み 1	デバッグ・ツールからアップロード(<u>U</u>)	(#(
実行を一瞬停 満	デバッグ・ツールから切断(<u>N</u>) Shift+F6	00
実行中に表示 表示更新間隔 🔳	停止(<u>S</u>) Shift+F5	1.J.C. 500
①デバッガをホットプラグイン準 「デバッグ」メニューの「ホットプラグイン」を選択し		
備状態へ移行	てください。	

・下記ダイアログが表示されることを確認します。





・E1 エミュレータとホットプラグアダプタ(ROE000010ACB00)との接続

E1 と ROE000010ACB00 を接続します。E1 エミュレータ側とホットプラグアダプタ側でコネクタをまっす ぐに接続してください。



・ホットプラグイン接続

接続後にデバッガの[OK]ボタンを押下してください。

	質問(Q0204001)	
?	ホット・ブラヴインの準備が完了しました。 パソコンとデバッグ・ターゲットを接続し、OKボタンを押してください。	
	① OK **ンセル	
プラグイン接続	[OK]ボタンを押下してくだ	さい。

[ホットプラグイン接続時に使用できない機能]

以下の機能はホットプラグイン接続時に使用できません。ホットプラグイン接続後に一度強制ブレーク をしてから使用してください。

・トレース機能

①ホッ

- ・端子マスク機能
- ・イベントブレーク機能/シーケンシャル・ブレーク機能
- ・ソフトウェアブレーク機能
- ・停止時のタイマ/シリアル系周辺エミュレーション動作/停止設定

2.2.5 デバッグ

- (1) メモリ表示更新機能
 - ・メモリパネルを開いてください。



①メモリパネルを開く	「表示」メニューを開いて[メモリ]→[メモリ 1]を選択 してください。
②RAM モニタ機能が動作している	メモリパネルの値がピンク色になり、定期的に表示が更
メモリパネルの確認	新されます。

- (2) グローバル変数の表示
 - ・ソース上から変数を登録してください。



①グローバル変数をウォッチパネルに	main.cソース内にある「g_i」をドラッグし、右ク
登録	リックしてください。その後「ウォッチ1に登録」
	を選択してください。
②ウォッチ登録されている変数の確認	変数の表示が定期的に更新されています。

- (3) 強制ブレーク
 - ・プログラムをブレークさせます。



- ①プログラムの強制ブレークメニュー上にある「停止」ボタンを押下してください。
- (4) ブレーク後のデバッグ機能
 - ・ソフトウェアブレーク

プログラム実行をした際、設定個所でブレークが発生します。

[™] →	Image: State of the state
①ソフトウェアブレークの設定	main.c ソース上にある「g_j++;」行のブレーク設定行を クリックしてください。
②ソフトウェアブレークの確認	プログラムを実行しソフトウェアブレークが発生するこ とを確認します。

・アクセスブレーク

変数に特定の値をWriteした場合、Readした場合、Write/Readした場合にブレークすることができる機能です。



①コンテキストメニューを開く	main.c ソース上で「g_i」をドラックし、右クリックし てください。
②アクセスブレークの設定	「ブレークの設定」→「書き込み合わせブレークを設定」 でブレークする値を指定します。任意の指定が可能で す。ここでは「0x01」を指定してください。
③アクセスブレークの確認	アクセスブレークが発生するとプログラムが止まりアク セスした場所と値が示されます。



2.2.6 ユーザシステムとの切断

(1) デバッグツールの切断

デバッグツールを切断してください。

- ※1 切断する場合は、プログラムを停止する必要があります。
- ※2 切断時にマイコンのリセットが発生します。デバッグツール切断後、エミュレータ接続中はリセット 状態が継続します。
- ※3 切断後、再度ホットプラグインする場合は、E1 をユーザシステムから外してマイコンを一度リセット してください。



①デバッグツールの切断

)切断 プログラムをブレークさせ、メニューにある「切断」ボ タンを押下してください。

(2) ユーザシステムの立ち下げ

・ユーザシステムの電源を落としてください。

- (3) E1 エミュレータの取り外し
 - ・E1 エミュレータとホスト PC 間の USB ケーブルを取り外します。
 - ・その後、E1 エミュレータとユーザシステムを取り外してください。
- (4) CS+の終了
 - ・CS+を終了してください。



3. 注意事項

3.1 デバッグ DTC 動作クロックに関する注意事項

デバッグ DTC の動作クロック供給設定は、通常の DTC と同じ DTCEN ビットで行います。このため、ホット プラグインおよび DTC 方式 RRM/DMM を使用する際に、DTCEN ビットを0 に設定しないようにしてください。

3.2 DTC の保留命令に関する注意事項

ホットプラグインおよび DTC 方式 RRM/DMM は DTC を使用して実現しているため、DTC 保留命令が連続する場合は DTC の起動が保留されます。DTC の起動が保留され続けるとホットプラグイン接続や RRM/DMM に失敗します。

[DTC 保留命令]

a) 無条件分岐命令 b) コールリターン命令

c)条件付分岐命令 d) コードフラッシュ領域へのリードアクセス命令

e) IFxx、MKxx、PRxx、PSW へのビット操作命令とオペランドに ES レジスタを含んだ 8 ビット操作命令 特に C 言語で無限ループ処理を行う場合、以下のようにアセンブラ展開されるので注意が必要です。



このような場合は無限ループ内にnop命令を使用するなどして該当命令が連続しないようにしてください。



3.3 データアクセスイベント関する注意事項

DTC 方式 RRM/DMM を使用する場合、RRM/DMM 対象の変数や SFR に対してデータアクセスイベントが設定されていると RRM/DMM のアクセスに対して以下のイベントが発生します。

・イベントブレーク(シーケンシャル含む)

・トレース開始/終了イベント

データアクセスイベントを変数/SFR に設定する場合は、同じ変数/SFR に対して DTC 方式 RRM/DMM を使用しないでください。

3.4 32bit 長以上の変数アクセスに関する注意事項

DTC 方式 RRM/DMM で使用している DTC のデータサイズは最大 16bit です。

このため、32bit 長以上の変数を読み出し/書き込みと、CPU による変数への書き込みが競合すると、誤った読み出し値や誤った書き込み値となることがあります。

例えば 32bit 長の変数を読み出す場合、上位 16bit と下位 16bit の読み出しの間に CPU によって変数に書 き込みが行われると、上位 16bit が書き込み前データで下位 16bit が書き込み後データとなった値を読み出 します。 3.5 スタンバイモードに関する注意事項

スタンバイモードに関して以下の注意事項があります。

・ホットプラグイン接続前

接続前に MCU がスタンバイモードに入っていて、低速オンチップオシレータと DTC の動作クロックのいずれかが停止している場合、ホットプラグイン接続ができません。

スタンバイモードに入っていることが原因でホットプラグイン接続できない場合はリトライ設定を活用 してください。

・ホットプラグイン接続後 ホットプラグイン接続後に STOP モードに入った場合、かつ DTC の動作クロックが高速オンチップオシレータでない場合、STOP モード中は DTC 方式 RRM/DMM を保留します。(STOP モード復帰後に再開します。)なお、デバッガはリトライ設定の「リトライ間隔」で指定した時間を DTC 方式 RRM/DMM の最大保留時間として扱っています。(リトライ間隔分 STOP モードから復帰しない場合、エラーとなります。)

3.6 リセットに関する注意事項

リセットに関して以下の注意事項があります。

・プログラム停止中の POR 回路によるリセット
 ホットプラグイン後、プログラム停止中に POR 回路によるリセットが発生すると、リセット発生から
 900ms 程度ユーザプログラムが実行後に停止します。
 ※ホットプラグイン用初期化プログラム実行完了後に停止可能となるため。

・プログラム実行中の POR 回路によるリセットと端子リセット
 ホットプラグイン後、プログラム実行中に POR 回路によるリセット、もしくは端子リセットが発生すると以下の設定が無効になります。
 ハードウェアブレーク設定(イベント無効)

- -トレース機能(イベント無効、トレース 0FF)
- -入力信号のマスク設定(マスク設定が全て 0FF)

また、ソフトウェアブレーク使用時に POR 回路によるリセット、もしくは端子リセットが発生し、再度 ホットプラグイン接続するまでの間にソフトウェアブレークを設定した命令を実行すると 0xFF コード 実行によるリセットが発生します。

なお、端子リセットに関しては、端子リセットのマスク機能をホットプラグイン接続してから強制ブレー ク後に使用することができます。このため、上記の動作が問題となる場合は端子リセットマスク機能を 使用してください。

・プログラム実行中の内部リセット(POR 回路によるリセットと端子リセットを除く) ホットプラグイン後、プログラム実行中に内部リセットが発生するとDTC 方式 RRM/DMM が中断されます。 この場合、リトライ間隔として設定された時間後に DTC 方式 RRM/DMM を再度開始します。

3.7 RAM の使用に関する注意事項

デバイス品種によってはホットプラグイン接続やDTC方式RRM/DMM、トレース機能を使用するためにRAM領域確 保が必要なものがあります。詳細は、各デバイスのユーザーズマニュアルハードウエア編(オンチップ・ト レース)をご参照ください。



ホームページとサポート窓口

ルネサス エレクトロニクスホームページ https://www.renesas.com

お問合せ先

https://www.renesas.com/jp/ja/contact-us

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

		改訂内容	
Rev.	発行日	ページ	ポイント
1.00	2013.03.29	—	初版発行
2.00	2013.08.20	2013.08.20 4 「ホットプラグイン接続後のデバッグ」追加	
		9	ホットプラグイン初期化関数を一部変更
			「初期化関数の使用方法」を追加
		13	「オンチップデバッグオプションバイト設定について」を追加
		18	ホットプラグインのリトライ設定に関する記載を追加
		20	「ホットプラグイン接続時に使用できない機能」を追加
		24	「ユーザシステムとの切断」を見直し
		25	「注意事項」を追加
3.00	2016.08.26	—	CubeSuite+を CS+に変更
		1	RL78/F15、 RL78/F1A を追加
		10	CS+ for CC用のホットプラグイン初期化関数を追加
4.00	2023.12.20	11	RL78/F2x 用のホットプラグイン初期化関数を追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテク ニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアース を施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いを してください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リ セット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセッ ト機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入によ り、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記 載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなって います。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識さ れて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後 に切り替えてください。リセット時、外部発振子(または外部発振回路)を用いたクロックで動作を開始するシステムでは、クロックが十分安定した 後、リセットを解除してください。また、プログラムの途中で外部発振子(または外部発振回路)を用いたクロックに切り替える場合は、切り替え先 のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_H (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_H (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

 リザーブアドレス(予約領域)のアクセス禁止
 リザーブアドレス(予約領域)のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス(予約領域) があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュ メモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合がありま す。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害(お客様または第三者いずれに生じた損害も含みます。以下同じです。)に関し、当社は、一切その責任を負いません。
- 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、 著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではあり ません。
- 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
- 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、 複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
- 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準: コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等 高品質水準:輸送機器(自動車、電車、船舶等)、交通制御(信号)、大規模通信機器、金融端末基幹システム、各種安全制御装置等 当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機 器・システム(生命維持装置、人体に埋め込み使用するもの等)、もしくは多大な物的損害を発生させるおそれのある機器・システム(宇宙機器と、 海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等)に使用されることを意図しておらず、これらの用途に 使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負い ません。

- 7. あらゆる半導体製品は、外部攻撃からの安全性を100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ 対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害(当社製品または当社製品が使用されているシス テムに対する不正アクセス・不正使用を含みますが、これに限りません。)から生じる責任を負うものではありません。当社は、当社製品または当社 製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為(「脆弱 性問題」といいます。)によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切 責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目 的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
- 8. 当社製品をご使用の際は、最新の製品情報(データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体 デバイスの使用上の一般的な注意事項」等)をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲 内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責 任を負いません。
- 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
- 10. 当社製品の環境適合性等の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合せください。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、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。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、一切その責任を負いません。
- 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
- 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
- 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に 支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

www.renesas.com

お問合せ窓口

ww

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口 に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の 商標です。すべての商標および登録商標は、それぞれの所有者に帰属し ます。